Martin Atzmueller
Dominik Benz
Andreas Hotho
Gerd Stumme (Eds.)

LWA 2010

Lernen, Wissen & Adaptivität
Workshop Proceedings
Kassel, 4.-6. Oktober 2010

Table of Contents

Preface

The joint workshop event LWA 2010 (Lernen, Wissen, Adaptivität) takes place from October 4th to October 6th 2010 in Kassel, Germany. Like in the years before the LWA hosts a broad scope of workshops of the special interest groups for

- Adaptivity and User Modeling (FG-ABIS)
- Information Retrieval (FG-IR)
- Knowledge Discovery and Machine Learning (FG-KDML)
- Knowledge and Experience Management (FG-WM)

In addition to the separate workshops of each special interest group we invited two talks covering current research questions in computer science:

- Jürgen Geck: Turning the Web Inside Out.
- Wolfgang Nejdl: Web of People – Improving Search on the Web.

We are grateful for the support of all members of the Knowledge and Data Engineering group at the University of Kassel, especially Monika Vopicka for handling many organizational and almost all issues with bureaucracy. Furthermore, we thank all participants of the workshops for their contributions. Additionally, we want to thank all reviewers for their careful help in selecting and improving the provided submissions.

Kassel, September 2010
Martin Atzmueller, Dominik Benz, Andreas Hotho, Gerd Stumme

Workshop on Knowledge Discovery and Machine Learning KDML 2010

Martin Atzmueller
Knowledge and Data Engineering Group
University of Kassel, Germany
atzmueller@cs.uni-kassel.de

Dominik Benz
Knowledge and Data Engineering Group
University of Kassel, Germany
benz@cs.uni-kassel.de

The KDML Workshop Series

The workshop *Knowledge Discovery, Data Mining, and Machine Learning (KDML) 2010* is organized annually by the Special Interest Group on Knowledge Discovery, Data Mining, and Machine Learning (FG-KDML, former FGML) of the German Society on Computer Science (GI).

The main goal of the workshop is to enable and stimulate the exchange of innovative ideas and to provide a forum for data mining and machine learning oriented researchers in order to discuss recent topics in these areas.

Submissions from current research out of these and adjacent areas were welcome. Moreover, contributions that describe work in progress or approaches that have not yet been investigated in depth yet were of special interest. As every year, the workshop was a forum for testing the viability of new ideas, by junior as well as senior researchers. Additionally, several resubmissions of previously published work proved to be interesting complements for initiating productive discussions.

The workshop is part of the workshop week *Learning - Knowledge Discovery - Adaptivity (LWA) 2010* that also features several other workshops. This provides the opportunity to meet researchers from the related special interest groups on Adaptivity and Interaction, on Information Retrieval, and on Knowledge Management, and fosters (inter-workshop) scientific discussions and the important exchange of ideas.

KDML 2010

The proceedings contain the papers presented at KDML 2010 held on October 4th–6th, 2010 in Kassel, Germany. 12 regular research papers, six short papers, and 12 resubmussions of previously published work, in addition to three posters, were accepted. The topics of interest of the KDML workshop series are:

- Mining and Analysis of Networks and Graphs
- Rule Learning
- Text Mining, Web Mining
- Distributed Data Mining
- Ubiquitous Knowledge Discovery
- Unsupervised and Semi-Supervised Learning
- Visual Analytics
- Knowledge Discovery in inductive databases
- Bioinformatics applications
- Data Stream Mining
- Temporal Knowledge Discovery

Workshop Chairs

- Martin Atzmueller, University of Kassel
- Dominik Benz, University of Kassel

Program Committee

- Martin Atzmueller, University of Kassel
- Dominik Benz, University of Kassel
- Stephan Doerfel, University of Kassel
- Folke Eisterlehner, University of Kassel
- Zeno Gantner, University of Hildesheim
- Robert Jäschke, University of Kassel
- Peter Kluegl, University of Würzburg
- Beate Krause, University of Würzburg
- Florian Lemmerich, University of Würzburg
- Björn-Elmar Macek, University of Kassel
- Leandro Marinho, Federal University of Maranhao
- Christoph Scholz, University of Kassel

We thank the authors for their submissions, and especially thank the Program Committee for their good work.

August 2010

Martin Atzmueller
Dominik Benz

Acknowledgements This volume has been produced in part using the EasyChair system[1]. We would like to express our gratitude to its author Andrei Voronkov.

[1] http://www.easychair.org

Lifted Conditioning for Pairwise Marginals and Beyond *

Babak Ahmadi and **Kristian Kersting** and **Fabian Hadiji**
Knowledge Dicovery Department, Fraunhofer IAIS
53754 Sankt Augustin, Germany
firstname.lastname@iais.fraunhofer.de

Abstract

Lifted belief propagation (LBP) can be extremely fast at computing approximate marginal probability distributions over single variables and neighboring ones in the underlying graphical model. It does, however, not prescribe a way to compute joint distributions over pairs, triples or k-tuples of distant random variables. In this paper, we present an algorithm, called *conditioned* LBP, for approximating these distributions. Essentially, we select variables one at a time for conditioning, running lifted belief propagation after each selection. This naive solution, however, recomputes the lifted network in each step from scratch, therefore often canceling the benefits of lifted inference. We show how to avoid this by efficiently computing the lifted network for each conditioning directly from the one already known for the single node marginals. This contribution advances the theoretical understanding of lifted inference but also allows one to efficiently solve many important AI tasks such as finding the MAP assignment, sequential forward sampling, parameter estimation, active learning, sensitivity analysis, to name only few. Our experimental results validate that significant efficiency gains are possible and illustrate the potential for second-order parameter estimation of Markov logic networks.

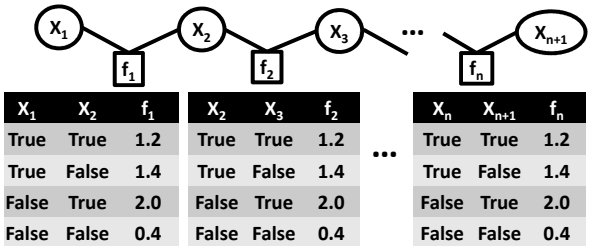
1 Introduction

There has been much recent interest in methods for performing lifted probabilistic inference, handling whole sets of indistinguishable objects together. Poole [2003], Braz *et al.* [2005], and Milch *et al.* [2008] have developed lifted versions of the variable elimination algorithm. Sen *et al.* [2009] presented a lifted variable elimination approach based on bisimulation. Most of these lifted inference approaches are extremely complex, so far do not easily scale to realistic domains and hence have only been applied to rather small artificial problems. A remarkable exception are lifted versions of belief propagation [Singla and Domingos, 2008; Kersting *et al.*, 2009]. Similar to Sen *et al.*'s approach, They grouped together random variables that have identical computation trees but now run a modified belief propagation (BP) on the resulting lifted, i.e., clustered network. Being instances of BP, they can be extremely fast at computing approximate marginal probability distributions over single variable nodes and neighboring ones in the underlying graphical model. Above all, they naturally scale to realistic domain sizes. Despite their success, however, lifted BP approaches do not provide a prescription to compute joint probabilities over pairs of non-neighboring variables in the graph. When the underlying graphical model is a tree, there is a single chain connecting any two nodes, and dynamic programming techniques might be developed for efficiently integrating out the internal variables. When cycles exist, however, it is not clear what approximate procedure is appropriate. The situation is even more frustrating when computing marginals over triples or k-tuples of distant nodes. As for the non-lifted case, sophisticated exact lifted inference algorithms are only tractable on rather small models and do not scale to realistic domain sizes. It is precisely this problem that we are addressing in this paper, that is we are interested in approximate lifted inference algorithms based on the conditioning idea that scale to realistic domain sizes. Specifically, we present *conditioned* LBP (CLBP), a scalable lifted inference algorithm for approximate inference based on conditioning. Essentially, we select variables one at a time for conditioning and run lifted belief propagation after each selection. This naive solution, however, recomputes the lifted network in each step from scratch, therefore often canceling the benefits of lifted inference. We show how to avoid this by efficiently computing the lifted network for each conditioning directly from the one already known for the single node marginals. There has been some prior work for related problems. Delcher *et al.* [1996] propose a data structure that allows efficient queries when new evidence is incorporated in singly connected Bayesian networks and Acar *et al.* [2008] present an algorithm to adapt the model to structural changes using an extension of Rake-and-Compress Trees. The only lifted inference approach we are aware of is the work by Nath and Domingos [2010] that was independently developed in parallel. The authors essentially simulate their lifting procedure for a set of changed variables, obtaining the adapted lifted network.

We also consider the problem of determining the best variable to condition on in each iteration to stay maximally lifted over all iterations and propose a simple heuristic. Our experimental evaluation including experiments on second-order parameter estimation for Markov logic networks [Richardson and Domingos, 2006] shows that significant efficiency gains are obtainable compared to naively running (lifted) BP in each iteration. CLBP may also have future applications in more advanced relational learning tasks such as active learning.

We proceed as follows. We start off by briefly reviewing LBP. Then, we introduce CLBP, prove its soundness, and

*This paper also appeared in PGM 2010



X_1	X_2	f_1
True	True	1.2
True	False	1.4
False	True	2.0
False	False	0.4

X_2	X_3	f_2
True	True	1.2
True	False	1.4
False	True	2.0
False	False	0.4

X_n	X_{n+1}	f_n
True	True	1.2
True	False	1.4
False	True	2.0
False	False	0.4

Figure 1: An example for a factor graph — a chain graph model with $n+1$ nodes — with associated potentials. Circles denote variables, squares denote factors.

touch upon the problem of determining the best variable to condition on at each level of recursion. Before concluding, we present the results of our experimental evaluation.

2 Lifted Belief Propagation

Let $\mathbf{X} = (X_1, X_2, \ldots, X_n)$ be a set of n discrete-valued random variables each having d states, and let x_i represent the possible realizations of random variable X_i. Graphical models compactly represent a joint distribution over $\mathbf{X}$ as a product of factors [Pearl, 1991], i.e.,

$$P(\mathbf{X} = \mathbf{x}) = Z^{-1} \prod_k f_k(\mathbf{x}_k) \ .$$

Each factor f_k is a non-negative function of a subset of the variables $\mathbf{x}_k$, and Z is a normalization constant. If $P(\mathbf{X} = \mathbf{x}) > 0$ for all joint configurations $\mathbf{x}$, the distribution can be equivalently represented as a log-linear model:

$$P(\mathbf{X} = \mathbf{x}) = Z^{-1} \exp\left[\sum_i w_i \cdot g_i(\mathbf{x})\right] ,$$

where the features $g_i(x)$ are arbitrary functions of (a subset of) the configuration $\mathbf{x}$.

Each graphical model can be represented as a factor graph. A factor graph, cf. Fig 1, is a bipartite graph that expresses the factorization structure of the joint distribution. It has a variable node (denoted as a circle) for each variable X_i, a factor node (denoted as a square) for each f_k, with an edge connecting variable node i to factor node k if and only if X_i is an argument of f_k. We assume one factor $f_i(\mathbf{x}) = \exp\left[w_i \cdot g_i(\mathbf{x})\right]$ per feature $g_i(\mathbf{x})$.

An important (#P-complete) inference task is to compute the conditional probability of variables given the values of some others, the evidence, by summing out the remaining variables. The belief propagation (BP) algorithm is an efficient way to solve this problem that is exact when the factor graph is a tree, but only approximate when the factor graph has cycles. Although this loopy BP has no guarantees of convergence or of giving the correct result, in practice it often does, and can be much more efficient than other methods. BP can be elegantly described in terms of sending messages within a factor graph. The message from a variable X to a factor f is

$$\mu_{X \to f}(x) = \prod_{h \in \mathrm{nb}(X) \setminus \{f\}} \mu_{h \to X}(x)$$

where $\mathrm{nb}(X)$ is the set of factors X appears in. The message from a factor to a variable is

$$\mu_{f \to X}(x) = \sum_{\neg\{X\}} \left(f(\mathbf{x}) \prod_{Y \in \mathrm{nb}(f) \setminus \{X\}} \mu_{Y \to f}(y) \right)$$

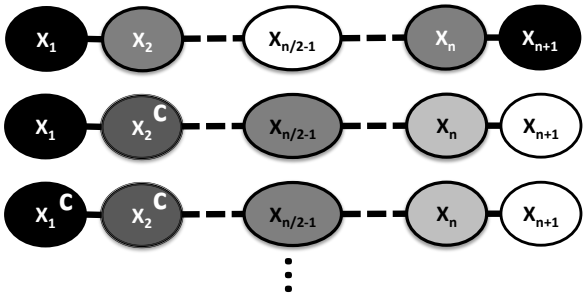


Figure 2: Supernodes, indicated by the shades of the original nodes, produced by repeatedly clamping nodes, indicated by "c",on a chain graph model with $n+1$ nodes. Factors have been omitted here. The conditioning order is $\pi = \{2, 1, 3, 4, \ldots, n-2, n-1, n+1, n\}$. After clamping X_2 all subsequent LBP runs work on the fully grounded network.

where $\mathrm{nb}(f)$ are the arguments of f, and the sum is over all of these except X, denoted as $\neg\{X\}$. The messages are usually initialized to 1, and the unnormalized belief of each variable X_i can be computed from the equation

$$b_i(x_i) = \prod_{f \in \mathrm{nb}(X_i)} \mu_{f \to X_i}(x_i) \ .$$

Evidence is incorporated by setting $f(\mathbf{x}) = 0$ for states $\mathbf{x}$ that are incompatible with it. Different schedules may be used for message-passing.

Although already quite efficient, many graphical models produce factor graphs with a lot of symmetries not reflected in the graphical structure. Consider the factor graph in Fig. 1. Although the factors involved are different on the surface, they actually share quite a lot of information, since the associated potentials are identical. Lifted BP (LBP) can make use of this fact. It essentially performs two steps: Given a factor graph G, it first computes a compressed factor graph $\mathfrak{G}$ and then runs a modified BP on $\mathfrak{G}$. We use fraktur letters such as $\mathfrak{G}$, $\mathfrak{X}$, and $\mathfrak{f}$ to denote the lifted, i.e., compressed graphs, nodes, and factors.

Step 1 of LBP — Lifting by Color-Passing (CP): Let G be a given factor graph with variable and factor nodes. Initially, all variable nodes fall into $d+1$ groups (one or more of these may be empty) — known states $s_1, \ldots, s_d$, and *unknown* — represented by colors. All factor nodes with the same associated potentials also fall into one group represented by a color. Now, each variable node sends a message to its neighboring factor nodes saying "I am of color C". A factor node sorts the incoming colors into a vector according to the order the variables appear in its arguments. The last entry of the vector is the factor node's own color. This color signature is sent back to the neighboring variables nodes, essentially saying "You have communicated with these kinds of nodes". The variable nodes stack the incoming signatures together and, hence, form unique signatures of their one-step message history. Variable nodes with the same stacked signatures are grouped together, and a new color is assigned to each group. The factors are grouped in a similar fashion based on the incoming color signatures of neighboring nodes. This CP process is iterated until no new colors are created anymore. As the effect of the evidence propagates through the factor graph, more groups are created. The final lifted graph $\mathfrak{G}$ is constructed by grouping nodes with the same color (signatures) into *supernodes* and all factors with the same color signatures into

superfactors. Supernodes (superfactors) are sets of nodes (factors) that send and receive the same messages at each step of carrying out BP and form a partition of the nodes in G. On this lifted network, LBP runs an efficient modified BP (MBP). We refer to [Singla and Domingos, 2008; Kersting *et al.*, 2009] for details.

(Step 2) Modified BP (MBP) on the Lifted Graph: The basic idea is to simulate BP carried out on G on $\mathfrak{G}$. [1] An edge from a superfactor $\mathfrak{f}$ to a supernode $\mathfrak{X}_i$ in $\mathfrak{G}$ essentially represents multiple edges in G. Let $c(\mathfrak{f}, \mathfrak{X}_i)$ be the number of identical messages that would be sent from the factors in the superfactor $\mathfrak{f}$ to each node in the supernode $\mathfrak{X}_i$ if BP was carried out on G. The message from a supervariable $\mathfrak{X}$ to a superfactor $\mathfrak{f}$ is $\mu_{\mathfrak{X}\to\mathfrak{f}}(x) =$

$$\mu_{\mathfrak{f}\to\mathfrak{X}}(x)^{c(\mathfrak{f},\mathfrak{X})-1} \cdot \prod_{\mathfrak{h}\in \mathrm{nb}(\mathfrak{X})\setminus\{\mathfrak{f}\}} \mu_{\mathfrak{h}\to\mathfrak{X}}(x)^{c(\mathfrak{h},\mathfrak{X})}$$

where $\mathrm{nb}(\mathfrak{X})$ now denotes the neighbor relation in the lifted graph $\mathfrak{G}$. The $c(\mathfrak{f}, \mathfrak{X}) - 1$ exponent reflects the fact that a subervariable's message to a superfactor excludes the corresponding factor's message to the variable if BP was carried out on G. Finally, the unnormalized belief of $\mathfrak{X}_i$, i.e., of any node X in $\mathfrak{X}_i$ can be computed from the equation

$$b_i(x_i) = \prod\nolimits_{\mathfrak{f}\in \mathrm{nb}(\mathfrak{X}_i)} \mu_{\mathfrak{f}\to\mathfrak{X}_i}(x_i)^{c(\mathfrak{f},\mathfrak{X})} \ .$$

Evidence is incorporated as in standard BP. by setting $\mathfrak{f}(\mathbf{x}) = 0$ for states $\mathbf{x}$ that are incompatible with it.

3 Lifted Conditioning

We are often faced with the problem of repeatedly answering slightly modified queries on the same network. Consider e.g. computing a joint distribution $P(X_1, X_2, \ldots, X_k)$ using LBP. A simple method is the following *conditioning* procedure that we call *conditioned* LBP (CLBP). Let π define a *conditioning order* on the nodes, i.e., a permutation on the set $\{1, 2, \ldots, k\}$ and its i-th element be denoted as $\pi(i)$. The simplest one is $\pi(i) = i$. Now, we select variables one at a time for conditioning, running LBP after each selection, and combine the resulting marginals. More precisely,

1. Run LBP to compute the prior distribution $P(X_{\pi(1)})$.
2. Clamp $X_{\pi(1)}$ to a specific state $x_{\pi(1)}$. Run LBP to compute the conditional distribution $P(X_{\pi(2)}|x_{\pi(1)})$.
3. Do this for all states of $X_{\pi(1)}$ to obtain all conditional distributions $P(X_{\pi(2)}|X_{\pi(1)})$. The joint distribution is now $P(X_{\pi(2)}, X_{\pi(1)}) = P(X_{\pi(2)}|X_{\pi(1)}) \cdot P(X_{\pi(1)})$.

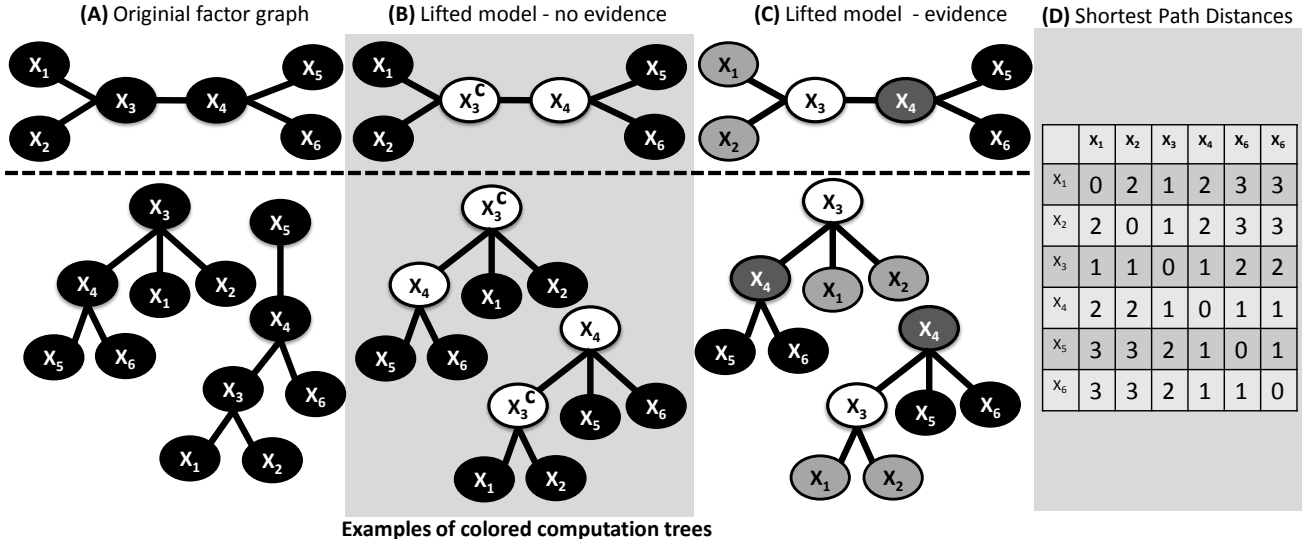
By iterating steps 2) and 3) and employing the chain rule we have $P(X_1, \ldots, X_k) = P(X_{\pi(1)}, \ldots, X_{\pi(k)}) = \prod_{i=1}^{k} P(X_{\pi(i)}|X_{\pi(i-1)}, \ldots, X_{\pi(1)})$. CLBP is simple and even exact for tree-structured models. Indeed, it is common to apply (L)BP to graphs with cycles as well. In this case the beliefs will in general not equal the true marginals, but often provide good approximations in practice. Moreover, Welling and Teh [2003] report that conditioning performs surprisingly well in terms of accuracy for estimating the covariance[2]. In the lifted case, however, the naive solution of repeatedly calling LBP may perform poorly in terms of running time. We are repeatedly answering slightly modified queries on the same graph. Because LBP generally lacks the opportunity of adaptively changing the lifted graph and using the updated lifted graph for efficient inference, it is doomed to lift the original model in each iteration again from scratch. Each CP run scales $\mathcal{O}(n \cdot m)$ where n is the number of nodes and m is the length of the longest path without loop. Hence, CLBP essentially spends $\mathcal{O}(k \cdot n \cdot m)$ time just on lifting. Moreover, in contrast to the propositional case, the conditioning order has an effect on the sizes of the lifted networks produced and, hence, the running time of MBP. It may even cancel out the benefit of lifted inference. Reconsider our chain example[3]. Figure 2 sketches the lifted networks produced over time when using the conditioning order $\pi = \{2, 1, 3, 4, \ldots, n-2, n-1, n+1, n\}$. That is, we clamp X_2, then all other nodes but X_n in ascending order. As one can see, clamping X_2 dooms all subsequent iterations to run MBP on the fully grounded network, canceling the benefits of lifted inference. In contrast, the order $\pi = \{1, n+1, 2, n, \ldots, n/2-1\}$ produces lifted and fully grounded networks alternatingly, the best we can achieve for chain models. We now address both issues.

Shortest-Paths Lifting: Consider the situation depicted in Fig. 3. Given the network in **(A)** and the prior lifted network, i.e., the lifted network when no evidence has been set **(B)**, we want to compute $P(X|x_3)$ as shown in **(C)**. To do so, it is useful to describe BP in terms of its *computation tree* (CT), see e.g. [Ihler *et al.*, 2005]. The CT is the unrolling of the (loopy) graph structure where each level i corresponds to the i-th iteration of message passing. Similarly we can view CP, i.e., the lifting procedure as a *colored computation tree* (CCT). More precisely, one considers for every node *X* the computation tree rooted in *X* but now each node in the tree is colored according to the nodes' initial colors, cf. Fig. 3**(bottom)**. Each CCT encodes the root nodes' local communication patterns that show all the colored paths along which node *X* communicates in the network. Consequently, CP groups nodes with respect to their CCTs: nodes having the same set of rooted paths of colors (node and factor names neglected) are clustered together. For instance, Fig. 3**(A)** shows the CCTs for X_3 and X_5. Because their set of paths are different, X_3 and X_5 are clustered into different supernodes as shown in Fig. 3**(B)**. The prior lifted network can be encoded as the vector $l = (0, 0, 1, 1, 0, 0)$ of node colors. Now, when we clamp a node, say X_3, to a value x_3, we change the communication pattern of every node having a path to X. Specifically, we change X_3's (and only X_3's) color in all CCTs X_3 is involved. This is illustrated in Fig. 3**(B)**. For the prior lifted network, the dark and light nodes in Fig. 3**(B)** exhibit the same communication pattern in the network. Consequently, X_3 appears at the same positions in all corresponding CCTs. When we now incorporate evidence on node X_3, we change its color in all CCTs as indicated by the "c" in Figs. 3**(B)** and **(C)**. This effects nodes X_1 and X_2 differently than X_4 respectively X_5 and X_6 for two reasons: (1) they have different communication patterns as they belong to different supernodes in the prior network; more importantly, (2) they have different paths connecting them to X_3

[1] For the sake of simplicity, we present simplified equations that neglect the positions a supernode may appear in a superfactor.

[2] The symmetrized estimate of the covariance matrix is typically not positive semi-definite and marginals computed from the joint distributions are often inconsistent with each other.

[3] When the graph is a chain or a tree there is a single chain connecting any two nodes and LBP together with dynamic programming can be used to efficiently integrate out the internal variables. When cycles exist, however, it is unclear what approximate procedure is appropriate.



	X_1	X_2	X_3	X_4	X_6	X_6
X_1	0	2	1	2	3	3
X_2	2	0	1	2	3	3
X_3	1	1	0	1	2	2
X_4	2	2	1	0	1	1
X_5	3	3	2	1	0	1
X_6	3	3	2	1	1	0

Figure 3: **(A):** Original factor graph. **(B):** Prior lifted network, i.e., lifted factor graph with no evidence. **(C):** Lifted factor graph when X_3 is set to some evidence. Factor graphs are shown **(top)** with corresponding colored computation trees **(bottom)**. For the sake of simplicity, we assume identical factors (omitted here). Ovals denote variables/nodes. The shades in **(B)** and **(C)** encode the supernodes. **(D):** Shortest-path distances of the nodes. The i-th row will be denoted d_i.

in their CCTs. The shortest path is the shortest sequence of factor colors connecting two nodes. Since we are not interested in the paths but whether the paths are identical or not, these sets might as well be represented as colors. Note that in Fig. 3 we assume identical factors for simplicity. Thus in this case path colors reduce to distances. In the general case, however, we compare the paths, i.e. the sequence of factor colors.

We only have to consider the vector d_3 of shortest-paths distances to X_3, cf. Fig. 3**(D)**, and refine the initial supernodes correspondingly. Recall that the prior lifted network can be encoded as the vector $l = (0, 0, 1, 1, 0, 0)$ of node colors. This is equivalent to (1) $l \oplus d_3$, the element-wise concatenation of two vectors, and (2) viewing each resulting number as a new color.

$$\begin{aligned}&(0,0,1,1,0,0) \oplus (1,1,0,1,2,2)\\ =_{(1)}\ &(01,01,10,11,02,02)\\ =_{(2)}\ &(0,0,1,2,3,3),\end{aligned}$$

the lifted network for $P(X|x_3)$ as shown in Fig. 3**(C)**. Thus, we can directly update the prior lifted network in linear time without taking the detour through running CP on the ground network. Now, let us compute the lifted network for $P(X|x_4, x_3)$. Essentially, we proceed as before: compute $l \oplus (d_3 \oplus d_4)$. However, the resulting network might be suboptimal. It assumes $x_3 \neq x_4$ and, hence, X_3 and X_4 cannot be in the same supernode. For $x_4 = x_3$, they could be placed in the same supernode, if they are in the same supernode in the prior network. This can be checked by $d_3 \odot d_4$, the element-wise sort of two vectors. In our case, this yields $l \oplus (d_3 \odot d_4) = l \oplus l = l$: the prior lifted network. In general, we compute $l \oplus (\bigoplus_s (\bigoplus_v d_{s,v}))$ where $d_{s,v} = \bigodot_{i \in s : x_i = v} d_i$, s and v are the supernodes and the truth value respectively. For an arbitrary network, however, the shortest paths might be identical although the nodes have to be split, i.e. they differ in a longer path, or in other words, the shortest paths of other nodes to the evidence node are different. Consequently we iteratively apply the shortest paths lifting. Let SN_S denote the supernodes given the set S as evidence. By applying the shortest path procedure we compute $SN_{\{X_1\}}$ from $SN_\emptyset$. This step might cause initial supernodes to be split into newly formed supernodes. To incorporate these changes in the network structure the shortest paths lifting procedure has to be iteratively applied. Thus in the next step we compute $SN_{\{X_1\} \cup \Gamma_{X_1}}$ from $SN_{\{X_1\}}$, where Γ_{X_1} denotes the changed supernodes of the previous step. This procedure is iteratively applied until no new supernodes are created. This essentially sketches the proof of the following theorem.

Theorem 1. *If the shortest-path colors among all nodes and the prior lifted network are given, computing the lifted network for $P(X|X_i, \ldots, X_1)$, $i > 0$, takes $\mathcal{O}(i \cdot n \cdot s)$, where n is the number of nodes, s is the number of supernodes. Running MBP produces the same results as running BP on the original model.*

Proof. For a Graph $G = (V, E)$, when we set new evidence for a node $X \in V$ then for all nodes within the network the color of node X in the $CCTs$ is changed. If two nodes Y_1, $Y_2 \in V$ were initially clustered together (denoted as $sn_0(Y_1) = sn_0(Y_2)$), i.e. they belong to the same supernode, they have to be split if the $CCTs$ differ. Now we have to consider two cases: If the difference in the $CCTs$ is in the shortest path connecting X with Y_1 and Y_2, respectively, then shortest-path lifting directly provides the new clustering. If the coloring along the shortest paths is identical the nodes' $CCTs$ might change in a longer path. Since $sn_0(Y_1) = sn_0(Y_2)$ there exists a mapping between the paths of the respective $CCTs$. In particular $\exists Z_1, Z_2$, s.t. $sn_0(Z_1) = sn_0(Z_2)$ from a different supernode, i.e. $sn_0(Z_i) \neq sn_0(Y_i)$, and $Y_1, \ldots, \underbrace{Z_1, \ldots, X}_{\Delta_1} \in CCT(Y_1)$, $Y_1, \ldots, \underbrace{Z_2, \ldots, X}_{\Delta_2} \in CCT(Y_2)$ and $\Delta_1 \in CCT(Z_1) \neq \Delta_2 \in CCT(Z_2)$ are the respective shortest paths for Z_1 and Z_2. Thus, by iteratively applying shortest-path lifting as explained above, the evidence propagates through and we obtain the new clustering. □

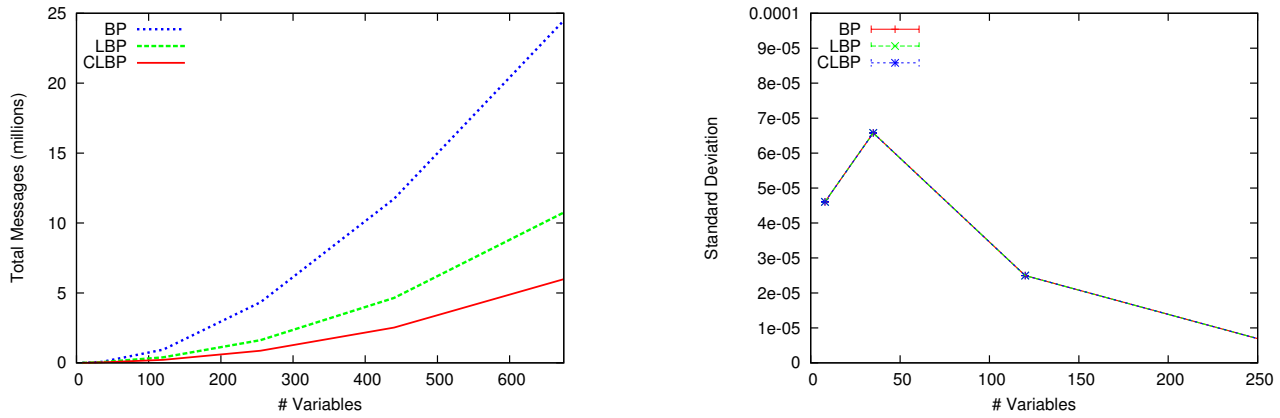


Figure 4: Pairwise Probability Estimates: **(Left)** Comparison of the total number of messages sent for BP, Lifted BP and "min-split" order CLBP for "Friends-and-Smokers" MLNs (including clustering messages for LBP and CLBP). **(Right)** The Standard Deviation of the error compared to the exact solution computed using the Junction Tree Algorithm.

On Finding a Conditioning Order: Clearly, CLBP will be most efficient for estimating the probability of a joint state when it produces the smallest lifted networks. This calls for the task of finding the most efficient[4] conditioning order. Here, we provide a generically applicable strategy based on the nodes' shortest-path colors to all other nodes. That is, in each conditioning iteration, we add that node having the smallest number of unique paths to all other nodes and, if possible, is a member of a supernode of one of the already clamped nodes. Intuitively, we select nodes that are expected to create the smallest number of splits of existing supernodes in each iteration. Therefore, we call it *min-split*. Although this increases the running time — each conditioning iteration now has an additional $\mathcal{O}(n^2)$ step — our experiments show that there are important cases such as computing pairwise joint marginals where the efficiency gains achievable due to a better lifting can compensate this overhead.

4 Experimental Evaluation

Our intention here is to illustrate the performance of CLBP compared to naively running LBP and BP. We implemented CLBP and its variants in Python and using LIBDAI library [Mooij, 2009] and evaluated the algorithms on a number of Markov logic networks.

In our first experiment, we compared CLBP to naively running LBP, i.e. lifting the network each time from scratch, and BP for computing pairwise probabilities. We generated the "Friends-and-Smokers" Markov logic network [Singla and Domingos, 2008] with 2, 5, 10, 15, 20, and 25 people, resulting in networks ranging from 8 to 675 nodes. The shortest-path lifting clearly pays out in terms of the total messages sent (including CP and shortest-path messages) as shown in Fig. 4 (**left**). Moreover, the accuracy estimates are surprisingly good and confirm Welling and Teh [2003]; Fig. 4 (**right**) shows the Standard Deviation of the difference compared to the exact solution computed using the Junction Tree (JT). The maximal error we got was below 10^{-4}. Note, however, that running JT with more than 20 persons was impossible due to memory and time restrictions.

In our second experiment we investigated CLBP for computing joint marginals. For the "Friends-and-Smokers" MLN with 20 people we randomly chose $1, 2, \ldots, 10$ "cancer" and "friends" nodes as query nodes. The joint state was randomly chosen. The results are averaged over 10 runs. Fig. 5 shows the cumulative number of messages (including CP messages). "Min-split" is indeed better. By chosing the order following our heuristic the cumulative number of supernodes and in turn messages is reduced compared to a random elimination order.

Finally, we learnt paramters for the "Friends-and-Smokers" MLN with 10 perople, maximizing the conditional marginal log-likelihood (CMLL). Therefore we sampled 5 data cases from the joint distribution. We compared conjugate gradient (CG) optimization using Polak-Ribiere with Newton conjugate gradient (NCG) optimization using the covariance matrix of MLN clauses computed using CLBP. The gradient was computed as described in [Richardson and Domingos, 2006] but normalized by the number of groundings of each clause. The results summarized in Fig. 6 confirm that information about dependencies among clauses is indeed useful: the second order method exhibits faster convergence.

5 Conclusion

We presented *conditioned lifted BP*, the first approach for computing arbitrary joint marginals using lifted BP. It relates conditioning to computing shortest-paths. Exploiting this link in order to establish runtime bounds is an interesting avenue for future work. By combining lifted BP and variable conditioning, it can readily be applied to models of realistic domain size. As our results show significant efficiency gains are obtainable, sometimes order of magnitude, compared to naively running (lifted) BP in each iteration. An interesting avenue for future work is to apply CLBP within important AI tasks such as finding the MAP assignment, sequential forward sampling, and structure learning. Furthermore, our results suggest to develop lifted cutset conditioning algorithms, see e.g. [Bidyuk and Dechter, 2007], and to lift Eaton and Ghahrmani's [2009] fast heuristic for selecting nodes to be clamped to improve CLBP's accuracy.

[4]This question is different from the more common question of finding highly accurate orders. The latter question is an active research area already for the ground case see e.g. [Eaton and Ghahramani, 2009], and is also related to the difficult question of convergent BP variants, see e.g. [Mooij *et al.*, 2007].

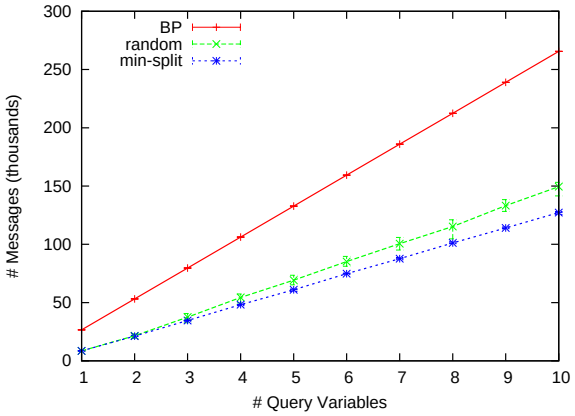


Figure 5: Number messages sent for computing joint marginals of varying size for BP, "random" and "min-split" order CLBP.

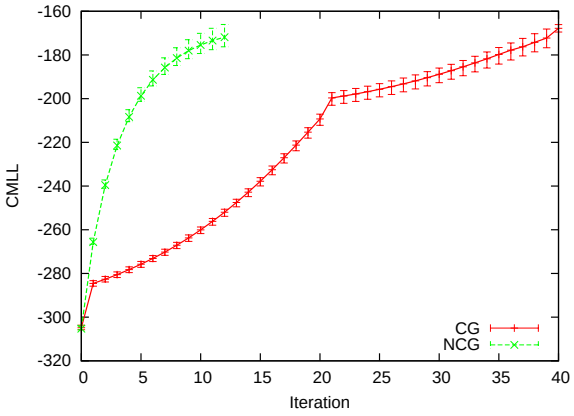


Figure 6: Learning curves for "Friends-and-Smokers" MLN. Optimization using clause covariances shows faster convergence.

Acknowledgements. This work was supported by the Fraunhofer ATTRACT fellowship STREAM and by the European Commission under contract number FP7-248258-First-MM.

References

[Acar *et al.*, 2008] U. A. Acar, A. T. Ihler, R. R. Mettu, and Ö. Sümer. Adaptive inference on general graphical models. In *Proc. of the Conf. on Uncertainty in Artificial Intelligence (UAI-08)*, 2008.

[Bidyuk and Dechter, 2007] B. Bidyuk and R. Dechter. Cutset sampling for bayesian networks. *Journal of Artificial Intelligence Research*, 28, 2007.

[de Salvo Braz *et al.*, 2005] R. de Salvo Braz, E. Amir, and D. Roth. Lifted First Order Probabilistic Inference. In *Proc. of the 19th International Joint Conference on Artificial Intelligence (IJCAI-05)*, pages 1319–1325, 2005.

[Delcher *et al.*, 1996] A. L. Delcher, A. J. Grove, S. Kasif, and J. Pearl. Logarithmic-time updates and queries in probabilistic networks. *JAIR*, 4:37–59, 1996.

[Eaton and Ghahramani, 2009] F. Eaton and Z. Ghahramani. Choosing a variable to clamp: Approximate inference using conditioned belief propagation. In *Proc. of the 12th International Conference on Artificial Intelligence and Statistics (AIStats-09)*, 2009.

[Ihler *et al.*, 2005] A.T. Ihler, J.W. Fisher III, and A.S. Willsky. Loopy belief propagation: Convergence and effects of message errors. *Journal of Machine Learning Research*, 6:905–936, 2005.

[Kersting *et al.*, 2009] K. Kersting, B. Ahmadi, and S. Natarajan. Counting belief propagation. In J. Bilmes A. Ng, editor, *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence (UAI–09)*, Montreal, Canada, June 18–21 2009.

[Milch *et al.*, 2008] B. Milch, L. Zettlemoyer, K. Kersting, M. Haimes, and L. Pack Kaelbling. Lifted Probabilistic Inference with Counting Formulas. In *Proc. of the 23rd AAAI Conf. on Artificial Intelligence (AAAI-08)*, July 13-17 2008.

[Mooij *et al.*, 2007] J. Mooij, B. Wemmenhove, H. Kappen, and T. Rizzo. Loop corrected belief propagation. In *Proc. of the 11th International Conference on Artificial Intelligence and Statistics (AIStats-09)*, 2007.

[Mooij, 2009] Joris M. Mooij. libDAI 0.2.3: A free/open source C++ library for Discrete Approximate Inference. http://www.libdai.org/, 2009.

[Murphy *et al.*, 1999] K.P. Murphy, Y. Weiss, and M.I. Jordan. Loopy Belief Propagation for Approximate Inference: An Empirical Study. In *Proc. of the Conf. on Uncertainty in Artificial Intelligence (UAI-99)*, pages 467–475, 1999.

[Nath and Domingos, 2010] A. Nath and P. Domingos. Efficient lifting for online probabilistic inference. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI-10)*, 2010.

[Park, 2002] J.D. Park. MAP complexity results and approximation methods. In *Proc. of the Conf. on Uncertainty in Artificial Intelligence (UAI-02)*, pages 388–396, 2002.

[Pearl, 1991] J. Pearl. *Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 2. edition, 1991.

[Poole, 2003] D. Poole. First-Order Probabilistic Inference. In *Proc. of the 18th International Joint Conference on Artificial Intelligence (IJCAI-05)*, pages 985–991, 2003.

[Richardson and Domingos, 2006] M. Richardson and P. Domingos. Markov Logic Networks. *Machine Learning Journal*, 62:107–136, 2006.

[Sen *et al.*, 2009] P. Sen, A. Deshpande, and L. Getoor. Bisimulation-based approximate lifted inference. In J. Bilmes A. Ng, editor, *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence (UAI–09)*, Montreal, Canada, June 18–21 2009.

[Singla and Domingos, 2008] P. Singla and P. Domingos. Lifted First-Order Belief Propagation. In *Proc. of the 23rd AAAI Conf. on Artificial Intelligence (AAAI-08)*, pages 1094–1099, July 13-17 2008.

[Welling and Teh, 2003] M. Welling and Y.W. Teh. Linear response for approximate inference. In *Proc. of NIPS-03*, pages 191–199, 2003.

The Smart-TSH-Finder: Crawling and Analyzing Tempo-Spatial Hotspots in Second Life

Akram Al-Kouz , Ernesto William De Luca, Jan Clausen, Sahin Albayrak
DAI-Labor, TU Berlin, 10587 Berlin, Ernst-Reuter-Platz
{Akram, ernesto.deloca, Jan.clausen, sahin.albayrak}@dai-labor.de

Abstract

In this paper we introduce the Smart Tempo-Spatial Hotspots Finder (Smart-TSH-Finder) for smart data crawling in the Second Life (SL) virtual world. Classical methods of crawling data from SL lead to irrelevant data content because of the dynamic nature of avatars and objects in SL. In order to build artificially intelligent expert avatar agents that are able to provide intelligent services for other typical avatars in virtual world we attempt to enhance the quality of extracted data from SL. Based on experimental observation, avatars tend to gather in some places for different amounts of time, which forms temporal and spatial hotspots. Utilizing the Tempo-Spatial characteristics of the avatars behavior in virtual worlds could improve the quality of the extracted data. Smart-TSH-Finder implements a Tempo-Spatial Hotspots finding mechanism to crawl dynamic contents such as chat conversations from Second Life. The system introduces two mechanisms: the Tempo-Spatial Hotspots Detection, and the Tempo-Spatial Hotspots Prediction. Our smart chat conversations that have been crawled showed good enhancement in content quality of the crawled chat conversation which will enrich the future textual analysis work. Additionally, we found that extracting avatars interactions and behavior in the Tempo-Spatial Hotspots in addition to chat conversations can help in generation a more coherent social network model for SL.

1 Introduction

Second Life[1] (SL) is a 3D virtual world developed by Linden Labs[2], a free client program called the Viewer enables its users to interact with each other through avatars. Avatars can explore, meet, socialize, participate in individual and group activities. SL has monthly unique users with repeat logins pick of 826,214 in March 2010; this represents 7% growth over the high set in Q4 2009, and 13% growth over March 2009[3]. The increasing popularity of SL as immersive virtual world made it an interesting target of research. In SL we have observed that avatars tend to gather in certain places (hotspots) in the land because of their sense to persistency. According to the operative definition of presence of [Vives and Slater, 2005], in that we suppose avatars will behave in the same way in SL. In order to build smart expert avatar agents able to provide smart services for other typical avatars in virtual world we need to extract dynamic data contents from SL. A data crawler capable to extract the desired data is needed. But SL is a commercial product, we do not have access to statistical data, and it is unlikely that such data would be made public by Linden Labs. On the other hand SL puts restrictions on the amount and the nature of data that can be crawled. The land in SL is divided into Parcels which are subdivided in a 256x256 meter region, content is only presented to users as they move close to the content location, thus, user can extract the public chat log within the silence cone of 20 meters, added to his own behavior and interactions with objects in SL, so the crawler must dynamically model the environment based on the previously mentioned restrictions.

We analyzed and utilized the Tempo-Spatial characteristics of the avatars behavior in SL to build Tempo-Spatial Hotspots finding system which contains smart agents (bots) able to extract dynamic contents such as chat conversations from intelligently detected or predicted hotspots in SL. Our goal was to determine trends in SL by analyzing the content of the extracted chat logs in future work to discover behavior patterns, conversation patterns, and conversation topics and to structure a social network in SL. Furthermore, evaluating how useful this information is in building user profile to be used in future work for recommendation or integration in our previous work done with the SpreeLand Expert Avatar [Al-Kouz et al., 2010]. We developed the proposed system based on the LibOpenMetaverse[4] open-source software related to the metaverse and virtual worlds. While our system is compatible with SL, it is also capable to integrate with other alternative virtual worlds in OpenSimulator[5].

In this work, we first present related work in the next section. Then, we introduce our system architecture and the related analysis mechanisms and algorithms used to build it, in the following section we discuss the empirical evaluation experiment. The paper concludes with a discussion and future work.

[1] www.secondlife.com
[2] http://lindenlab.com
[3] http://blogs.secondlife.com/community/blog/2010/04/28/economy-hits-new-all-time-high-in-q1-2010
[4] LibOpenMetaverse http://www.libsecondlife.org
[5] http://opensimulator.org/wiki/Main_Page

2 Related Works

At the moment, there is no previous work to smart use of the Tempo-Spatial characteristics of the avatars behavior in SL and related virtual worlds for enhancing the quality of crawled data. However, there are some published studies about utilizing both of the characteristics separately. Due to the lack of publicly chat data sets and the closed nature of SL as commercial product, the field has not yet gained great interest as opposed to the currently prevailing research on the typical Instant Messaging. Obviously, some research presents an intelligent agent crawler designed to collect user-generated content in SL. The agents navigate autonomously through the world to discover regions. It shows that virtual worlds can be effectively extracted using autonomous agent crawlers that emulate normal user behavior [Eno et al., 2009]. A refined approach presented analysis and visualization of the spatial and temporal distribution of user interaction data collected in 3D virtual worlds [Börner et al., 2001]. However it is not discussing its effect on the extracted data quality. A more sophisticated model developed a chat message analysis system called IMAnalysis using text mining techniques, which includes functions for chat message retrieval in general and not dedicated to 3D virtual worlds [Hui et al., 2008]. There has been some work on crawling and exploring virtual worlds for reasons other than content collection [Varvello et al., 2008], [La and Pietro, 2008]. Even though, none of mentioned researches used Tempo-Spatial characteristics of SL to enhance the quality of crawled data.

3 Our Approach

Based on our observation, SL avatars tend to gather in specific trendy spots for variant amount of time, taking this fact into consideration adding to the constraints on data crawling by Linden Labs, we find that the classical methods of crawling data from SL such as extracting data by a logged in automated bot into SL in specific location, or by making the bot to follow some avatars to imitate the natural behavior of the avatars, lead to irrelevant data content, because of the dynamic nature of avatars and objects in SL. In our attempt to enhance the quality of extracted data from SL, a smart data crawler system based on the Tempo-Spatial analysis of the avatars behavior was developed to crawl dynamic contents such as chat conversations from SL. Our system is based on the client/server protocol of the open source LibOpenMetaverse library.

3.1 System Architecture

As we have shown in Figure 1, the architecture of our proposed Smart-TSH-Finder system has two main modules: the Virtual Environment Layer module and the Smart-TSH-Finder Crawler module. The first establishes the network communications with the virtual world. The last contains three sub modules: Login Manager, Heat Maps Manager, and Crawling Manager. Heat Maps Manager introduces two alternative smart mechanisms to find the trendiest hotspots in the virtual world suitable to crawl dynamic contents from: the Tempo-Spatial Hot Spots Detection service, and the Tempo-Spatial Hot Spots Prediction service.

The Virtual Environment Layer Module

The Virtual Environment Layer Module is an intermediate network abstraction layer between Smart-TSH-Finder Crawler and the virtual world, it was built on top of LibOpenMetaverse library to customize some of the functionalities required by our Smart-TSH-Finder Crawler. LibOpenMetaverse runs on top of the .NET runtime and maintains compatibility with the Second Life protocol and can be used for creating clients and automatons in Second Life, OpenSimulator or other virtual worlds which use the Second Life Protocol.

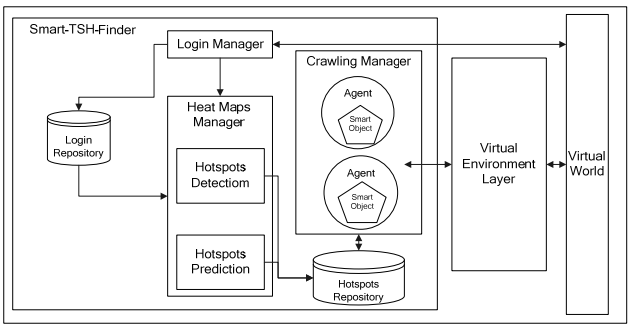


Figure1: Smart-TSH-Finder Architecture

The Smart-TSH-Finder Module

The Smart-TSH-Finder Module is a distributed artificial intelligence agent built on top the Virtual Environment. It is capable to connect multiple lands and extract dynamic contents such as chat conversations logs and some avatars' behavior actions. Smart-TSH-Finder Crawler has three sub modules: the Login Manager sub module, the Heat Maps Manager sub module and the Crawling Manager sub module.

The Login Manager Sub Module

The Login Manager is capable to connect to multiple lands simultaneously, it handles the authentication processes with the virtual environment, an active SL user credentials are required. After login it stores all the standard retrieved data from the virtual world in the Login Repository as explained in Figure 1.

The Heat Maps Manager Sub Module

The Heat Maps Manager is the core of our system, it uses the data stored in the Login Repository to generate a spatial heat map for the currently logged in land as shown in Figure 2, taking into consideration the virtual world specifications, such as the size of the Parcel (256*256 meters) and silence cone (20 meters), it generates a 2D matrix corresponding to the spatially discovered hotspots. The generated 2D matrix is used as the base for the smart Tempo-Spatial analysis in the Detection Service and the Prediction Service to determine the trendiest spots that are suitable to extract informative data from.

Detection Service

The Detection Service implements the Detection Algorithm which will be discussed in the first part of the Smart-TSH-Finder Algorithms section to discover the trendy Tempo-Spatial spots. It will suggest the most fitting spots between temporal and spatial spots, added to that it has the capability to use a baized criteria specified by the user to retrieve hotspots temporally or spatially baized. The retrieved hotspots will be stored in the Hotspots Repository.

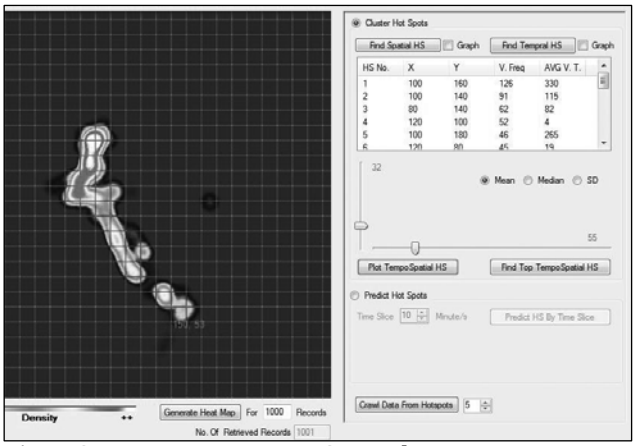


Figure 2: Heat Maps Manager Screenshot.

Prediction Service

The Prediction Service implements the Prediction Algorithm which will be discussed in the second part of the Smart-TSH-Finder Algorithms section to predict the next trendy spatial spots based on time stamp. It will predict the trendiest spots based on a user selected time slice, and predict the next probable hotspot using linear regression. The predicted hotspots will be sorted in the Hotspots Repository.

Crawling Manager Sub Module

The Crawling Manager is responsible for management of the data crawling process. It uses the data stored in the Hotspots Repository to generate a smart agent for each hotspot, the generated smart agent is programmatically controlled bot, which has the capability to extract dynamic contents from the virtual world. Furthermore, it can handle some Smart Objects. Smart Object is a geometric object, such as cube or circle, manually prebuilt in a free land and attached to the user's avatar, it has script able to response to some interactions and extracts some data. The extracted data will be stored in the Hotspots Repository for further analysis.

3.2 Smart-TSH-Finder Algorithms

In this section we describe the algorithms used to analyze the Tempo-Spatial behavior of the avatars based on the spatial heat map for the currently logged in land generated by the Heat Maps Manager as shown in Figure 2. The corresponding 2D matrix will be used as the base for the smart Tempo-Spatial analysis mentioned in the Detection mechanism and the prediction mechanism to determine the hotspots that are suitable to extract data from.

Hotspots Detection Algorithm

In the following we present an algorithm by which we detect hotspots of the land. The detection algorithm is an unsupervised learning method to assign set of Tempo-Spatial observations which are the visits frequency of the location as spatial factor and the average visit time of the location as temporal factor. In our experiment of the discovered temporal and spatial hotspots exhibit a Pearson correlation coefficient of approximately 0.74, details are provided by Figure 3, where the x axis is a numeric representation of the hotspot, the left y axis represents the visits frequency and the right y axis represents the average visit time in second, the plotted curve with diamonds represents the spatial values and the curve with circles represent the temporal values.

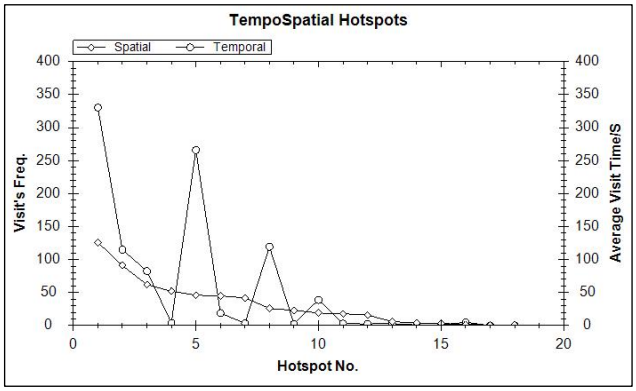


Figure 3: Tempo-Spatial Relation of Hotspots.

Following we describe how Tempo-Spatial Hotspots are discovered. First the mean or median respectively for each coordinate is computed. If a spot has a value greater than the mean in each component, it is considered as a Tempo-Spatial Hotspot. By this we find the most suitable fitting between the temporal and spatial hotspots. Furthermore, the Detection Service has alternative statistical techniques to find fitting between the temporal and spatial hotspots leading to discovering the most trendy hotspots both temporally and spatially as shown in Figure 4, where the x axis is the average visit time per hotspot and the y axis is the visits frequency of that hotspot, the fitted hotspots are plotted as big circles, the others as small diamonds.

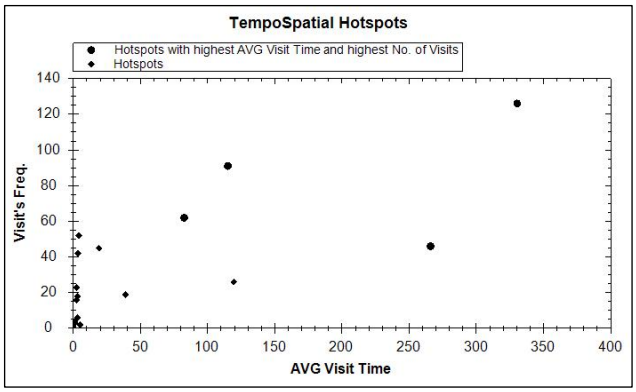


Figure 4: Tempo-Spatial Fitted Hotspots.

The default fitting criteria is the logical AND between the mean of the temporal hotspots and the mean of spatial hotspots, further the user can alternate between median and standard deviation as fitting criteria, fitted hotspots should be greater than the components means, median, or standard deviation respectively. On the other hand, the Detection Service will enable the user of the system to specify the biases into the temporal direction to crawl long chat conversations with low number of participants, or into the spatial direction to crawl long chat conversations with high number of participants.

Hotspots Prediction Algorithm

The Hotspots Prediction Algorithm uses the previously generated Heat Map to predict the coming hotspots in terms of time. It uses the linear regression analysis to ex-

pect the most top hotspots and its sequence in the next time slice. The spatial hotspots will be classified based on the user selected time slice generating a 2D matrix HS [i, j]. The first index i is a representation of the hotspot number, the second index j is a representation of the accumulated time slices and the value at HS [i, j] is the visit frequency of the ith hotspot at the jth time slice as shown in the Table 5.

HS/Time	*i*1	*i*2	*i*3	*i*4	*i*5	*i*6	*i*7
***j*1**	124	24	75	2	4	133	41
***j*2**	93	53	55	4	3	30	8
***j*3**	24	45	2	3	12	33	17
***j*4**	53	13	12	0	0	18	0
***j*5**	0	10	22	0	5	10	20
***j*6**	0	0	0	0	12	12	0
***j*7**	0	0	5	0	0	0	12

Table 5: Sample Hotspots Matrix

The Hotspots Prediction Service generates a 3D surface representation of HS matrix, as shown in Figure 6 hotspots reach their peak at different times. The system use the current time stamp to predict the next trendy spot by retrieving the correspondence visit frequency of all the hotspots in that time interval into two vectors, one to store the hotspot number and the other to store the corresponding visit frequency. The Linear Regression algorithm will be applied on the two vectors, hotspot with the least absolute value of residuals will be suggested as the next hotspot.

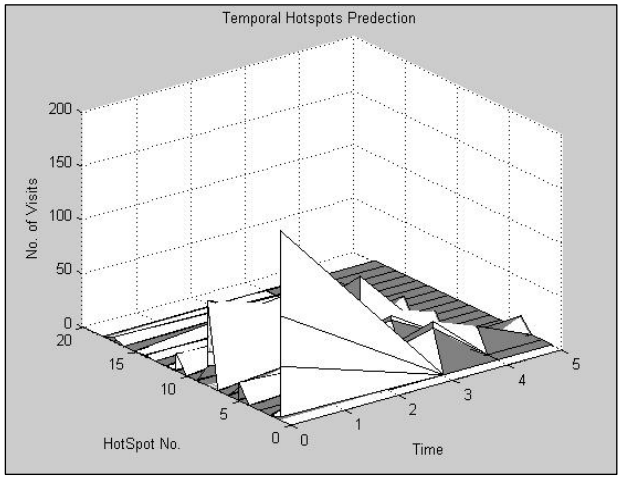


Figure 6: 3D Representation of the Sample Hotspots Matrix in Table 4.

4 Evaluation

In order to evaluate our system, we compare the quality of the extracted data in both the classical methods and in our proposed method based on the mentioned Tempo-Spatial Hotspots Finding Algorithms.

4.1 Data Crawling

We used Solace Cove Land[6] to do the experiment, Solace Cove is a popular Second Life virtual space for meeting friends, doing business, and sharing knowledge. We ran our system in the mentioned land five times for thirty minutes every time. Our system found the Tempo-Spatial Hotspots and sent four smart agents to that spots to start data extraction. At the same time we logged a bot in Solace Cove Land using classical LibObenMetaverse dashboard client and start extracting data by enforcing the bot to follow a specific avatar and record the chat log. In the previously mentioned two cases we fired up a specific discussion question "How was the FIFA world cup 2010 in South Africa" several times, and then we extracted avatars responses for thirty minutes. We got four chat log files from our system prefixes with hotspot number each one contains a different number of utterances and one chat log file from the classical dashboard prefixes with "DB" as shown in Table 7.

4.2 Data Preparation

In order to compare the quality of the crawled data from our system with the quality of the crawled data from the classical dashboard, we applied a windowing mechanism to split chat log files into smaller files. We considered the window size of eight utterances, six utterances as the average size of chat conversation based on [Ogura *et al.*, 2004], one for pre overlapping and one for post overlapping. Utterances fit in each window are stored as a separate file prefixes with chat log file name and window number as in Table 7.

Chat Log Files	No. of Utterances	No. Of Windows
HS1ChatLog	213	35
HS2ChatLog	322	53
HS3ChatLog	105	17
HS4ChatLog	96	16
DBChatLog	386	64

Table 7: Sample Extracted Chat Log files and its Windows

4.3 Evaluation Process

In our evaluation process we considered the previously fired question "How was the FIFA world cup 2010 in South Africa" as query, and we used a ranking function based on summing the TF–IDF[7] weight (term frequency–inverse document frequency) for each query term to evaluate how important that term is to a document in a collection or corpus. In this case each window file considered as a document and all the windows corresponding to one chat log file considered as corpus. To normalize the results we evaluated the first windows as the number of windows in chat log file of the lowest number of windows. We used the algorithm in Figure 8 to rank each chat log file. Chat log files with higher rank are the most relevant to our query.

4.4 Evaluation Results

We conducted five runs of our experiment and took the average results to generalize observations. Average results show that three out of the four hotspots chat log files (HS1ChatLog, HS3ChatLog and HS4ChatLog) which were extracted by our system have higher rank than the dashboard chat log file "DBChatLog" which extracted by the classical method. Thus, chat log file "HS2ChatLog"

6 http://maps.secondlife.com/secondlife/Solace20Cove/121/81/24

7 http://nlp.stanford.edu/IR-book/html/htmledition/tf-idf-weighting-1.html

has rank lass than but close to the rank of dashboard chat log file "DBChatLog". Further analysis of the results and extracted data has been done to discover that dashboard chat log file "DBChatLog" beyond chat log file "HS2ChatLog" in rank because the followed avatar used in the classical way hanged in one place for long time and that place was selected by our system as a hotspot, so our extracted data at this hotspot is a subset of the "DBChat-Log" file which explains why we got low rank in this hot-spot. Beyond that our system showed high ranking for the extracted data from the discovered hotspots which means our extracted data is more relevant and coherent.

For each chat log file
  For each window
    ChatLogFileRank += *WindowRank*

$$WindowRank = \sum_{iTerm}^{nTerm} TF \times IDF$$

Terms= "word", "cup", "2010","south","africa"
iTerm: First entry in Terms
nTerm: Last entry in Terms

$$TF = \frac{TT}{TW}$$

TT: No. of times term appears in window
TW: No. of words in window

$$IDF = Log\left(\frac{W}{WT}\right)$$

W: No. of windows in chat log file
WT: No. of the windows the term appeared in

Figure 8: Chat Log File Ranking Algorithm

5 Conclusion and Future Work

A Smart-TSH-Finder for Second Life has been developed as a substitute to the classical methods of crawling data from SL. The proposed system enhance the quality of the extracted data, which considered a major step in building artificially intelligent expert avatar agents able to provide smart services for other typical avatars in virtual world. Smart-TSH-Finder Crawler utilized the Tempo-Spatial characteristics of the avatars behavior in virtual worlds to improve the quality of the extracted data to discover the temporal and spatial hotspots that avatars tend to gather in. The system introduces two mechanisms: the Tempo-Spatial Hotspots Detection, and the Tempo-Spatial Hot-spots Prediction. System user have the choice to use the former or the last mechanism to determine the most trendy hotspots, after that the system creates an agent for each trendy hotspot and send it to that hotspot, the generated agent starts crawling data from its desired hotspot.

The empirical evaluation experiment of the crawled chat conversations showed good enhancement in content quality which enriches the future textual analysis work to discover behavioral patterns, conversation patterns, and conversation topics. Additionally, we found that extract-ing avatars interactions and behavior in the Tempo-Spatial Hotspots in addition to chat conversations can help in modeling a social network for Second Life. Furthermore, evaluating how useful this information is in building user profile to be used in future work for recommendation or integration with our previous work done with our previous work.

References

[Vives and Slater, 2005] Sanchez Vives, M.V., Slater, M, From presence to consciousness through virtual reality. Nature Reviews Neuroscience 6(4), 332–339, 2005.

[Al-Kouz *et al.*, 2010] Akram Al-Kouz, Winfried Um-brath, Ernesto William De Luca, Sahin Albayrak, SpreeLand: An Expert Avatar Solution for Virtual World E-Learning Environments, the International Conference on E-Learning in the Workplace, 2010.

[Eno *et al.*, 2009] Joshua Eno, Susan Gauch, Craig Thompson, Intelligent Crawling in Virtual Worlds IEEE/WIC/ACM International Conference on Web In-telligence and Intelligent Agent Technology – Work-shops, 2009.

[Börner *et al.*, 2001] Katy Börner, William R. Hazlewood & Sy-Miaw Lin, Visualizing the Spatial and Temporal Distribution of User Interaction Data Collected in Three-Dimensional Virtual Worlds, 2001.

[Hui *et al.*, 2008] Hui, S.C. , Yulan He , Haichao Dong , Text Mining for Chat Message Analysis, IEEE Confer-ence on Cybernetics and Intelligent Systems, 2008.

[Varvello *et al.*, 2008] M. Varvello, F. Picconi, C. Diot, and E. Bier sack, Is there life in Second Life?, Thom-son Technical Report CR-PRL-2008-07-0002, 2008.

[La and Pietro, 2008] C-A. La and M. Pietro, Characteriz-ing user mobility in Second Life, Technical Report RR-08-212, Institut Eurecom, 2008.

[Ogura *et al.*, 2004] Kanayo Ogura, Masato Ishizaki and Kazushi Nishimoto, A Method of Extracting Topic Threads Towards Facilitating Knowledge Creation in Chat Conversations, Knowledge-Based Intelligent In-formation and Engineering Systems, 2004.

Stream-based Community Discovery via Relational Hypergraph Factorization on Evolving Networks

Christian Bockermann and **Felix Jungermann**
Technical University of Dortmund,
Artificial Intelligence Group
Baroper Strasse 301, Dortmund, Germany
{bockermann,jungermann}@ls8.cs.tu-dortmund.de

Abstract

The discovery of communities or interrelations in social networks has become an important area of research. The increasing amount of information available in these networks and its decreasing life-time poses tight constraints on the information processing – storage of the data is often prohibited due to its sheer volume.

In this paper we adapt a flexible approach for community discovery offering the integration of new information into the model. The continuous integration is combined with a time-based weighting of the data allowing for disposing obsolete information from the model building process.

We demonstrate the usefulness of our approach by applying it on the popular *Twitter* network. The proposed solution can be directly fed with streaming data from *Twitter*, providing an up-to-date community model.

1 Introduction

Social networks like *Twitter* or *Facebook* have recently gained a lot of interest in data analysis. A social network basically consists of various types of entities – such as *users*, *keywords* or *resources* – which are in some way related to one another. A central question is often the discovery of groups of individuals within such networks - the finding of *communities*. Thus, we are seeking for a clustering of the set of entities into subsets where the individuals within each subset are most similar to each other and are most dissimilar to the entities of all other subsets. The similarity of entities is provided by their relations to one another.

The relations between different entities are implied by the communication taking place within the network. Users exchange messages, which contain references to other users, are tagged with keywords or link to external resources by means of URLs. Figure 1 shows a message from the *Twitter* platform, which implies relations between the user `yarapavan`, the URL `http://j.mp/fpga-mr` and the tag `#ML`.

A natural perception of a social network is that of a connected graph, which models each entity as a node and contains (weighted) edges between related entities. Such a graph can be easily described by its adjacency matrix: with d being the number of entities in our social network, we will end up with a (sparse) matrix $\boldsymbol{A}$ of size d^2, where $\boldsymbol{A}_{i,j} = w$ if entity i is related to j with weight w and 0 otherwise. However this representation is not well-suited for n-ary relations.

Figure 1: Example tweets of the *Twitter* platform

A well-established representation of multi-dimensional relations is given by tensors [1; 2; 5; 17; 12; 6; 19]. A tensor is a multi-way array and can be seen as a generalization of a matrix. Tensors have been successfully used in multi-dimensional analysis and recently gained attention in social network mining [1; 2; 5]. In the case of social networks, tensors can be used to describe n-ary relations by using one tensor for each type of relations. Ternary relations of type (*user*,*tag*,*url*) can then be described by a mode-3 tensor $\boldsymbol{\mathcal{X}}$ with

$$\boldsymbol{\mathcal{X}}_{i,j,k} = \begin{cases} w & \text{if user } i\text{, tag } j \text{ and url } k \text{ are related} \\ 0 & \text{otherwise.} \end{cases}$$

More complex n-ary relations will be reflected in tensors of mode-n.

Tensor based Community Discovery

Community discovery in such tensor representations is mapped to a decomposition of the tensors into a product of matrices $\boldsymbol{U}^{(i)} \in \mathbb{R}^{m_i \times k}$ which approximates the tensor

$$\boldsymbol{\mathcal{X}} \approx [z] \prod_i \times_{d_i} \boldsymbol{U}^{(i)}.$$

Each of the matrices $\boldsymbol{U}^{(i)}$ in turn reflects a mapping of entities to clusters $\{1, \ldots, k\}$. The $[z]$ factor is a super-diagonal tensor which serves as a "glue element" – see Section 3 for details. A variety of different decomposition techniques such as Tucker3 or PARAFAC (CP) has been previously proposed [3; 14; 7]. Approximation is commonly measured by some divergence function. In [5] the authors proposed a clustering framework based on tensor decompositions which has been generalized for Bregman divergences. In [4] Bader et al. used CP tensor decomposition to detect and track informative discussions from the Enron email dataset by working on the ternary relation (*term*,*author*,*time*). These approaches have been applied to decompose single tensors. In [16] the authors introduced METAFAC, which is a factorization of a *set of tensors* with shared factors ($\boldsymbol{U}^{(q)}$ matrices). This allows for the discovery of one global clustering based on multiple tensor descriptions of the data. The time complexity for these tensor

decompositions is generally given by the number of non-zero elements of the tensors (provided that a sparse representation is used).

Stream-based Community Discovery
The majority of the tensor decomposition methods so far is based on a static data set. To incorporate streaming data, the stream is broken down into blocks and the decompositions are re-computed for each of the new blocks [16]. A common way to handle time is to introduce a trade-off factor of the old data and the data contained in the new blocks.

In [18] Sun et al. presented dynamic tensor analysis. They handle n-ary relations by tensor decomposition using stream-based approximations of correlation matrices. They also presented a stream-based approach which is not really comparable to ours. They are processing a tensor containing data by unfolding the tensors to every single mode and after that they are handling every column of the resulting matrices in a stream to update their model. In reality, we cannot assume such an original tensor to be given. In contrast to [18], we consider multiple relations which have to be updated at each iteration instead of just one.

Contributions

The critical bottle-neck within the tensor decomposition methods often is their runtime. As of [16], the runtime for a decomposition of a set of tensors can be bound by $O(N)$, where N is the number of entries in all tensors. However, this number can be rather large – we extracted about 590k entries (relations) from 200k messages of the *Twitter* platform.

In this work, we present an adaption of the METAFAC framework proposed in [16]. Our contributions are as follows:

1. We integrate a sampling strategy into the METAFAC framework. Effectively we limit the maximum size of the tensors – and therefore N – and use a least-recently-used approach to replace old entities if the limit of an entity type exceeds.
2. We introduce a time-based weighting for relations contained within the tensors. These weights will decrease over time, reflecting the decreasing importance of links within the social networks.
3. We present an adaption of the METAFAC factorization which allows for a *continuous integration* of new relations into the factorization model. Instead of running the optimization in a per-block mode, we provide a way to simultaneously optimize the model while new data arrives.
4. Finally, we provide an evaluation of our proposed adaptions on real-world data.

The rest of this paper is structured as follows: Section 2 formalizes the problem and presents the METAFAC approach on which this work is based. Following that, we give an overview of tensor decomposition in Section 3 and provide the basics for the multilinear algebra terminology required. In Section 4 we introduce our stream-based adaption of the METAFAC algorithm. We evaluated our streaming approach on real world data (Section 5) and present our findings in Section 6.

2 Multi-Relational Graphs

As denoted above, a social network generally consists of a set of related entities. In general, we are given sets $V_1, \ldots, V_k$ of entities of different types, such as *users*, *keywords* or *urls*. Let V_i be the i-th type of entities, e.g. V_1 corresponds to *users*, V_2 refers to *keywords* and so on. A *relation* then is a tuple of entities, e.g. a *user-keyword* relation (u_1, k_1) is an element of $V_1 \times V_2$. We also refer to $R := V_1 \times V_2$ as the *relation type* R of the relation (u_1, k_1).

The entities are given as strings, and we define a mapping φ_i for each entity type V_i, which maps entities to integers

$$\varphi_i : V_i \to \{0, \ldots, |V_i| - 1\}.$$

The mapping φ_i can be some arbitrary bijective function. For some $w \in V_i$ we refer to $\varphi_i(w)$ as the index of w. We denote the string of an entity given by its index j by $\varphi^{-1}(j)$. This allows us to identify each entity by its index and enables us to describe a set of relations between entities by a tensor.

A tensor $\boldsymbol{\mathcal{X}}$ is a generalization of a matrix and can be seen as a higher-order matrix. A mode-k tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times \ldots \times I_k}$ is a schema with k dimensions where

$$\boldsymbol{\mathcal{X}}_{i_1,\ldots,i_k} \in \mathbb{R}, i_j \in I_j$$

denotes the entry at position $(i_1, \ldots, i_k)$. For $k = 2$ this directly corresponds to a simple matrix whereas $k = 3$ is a cube.

With the mappings φ_i of entities and the tensor schema, a set of relations $X \subseteq V_{i_1} \times \ldots \times V_{i_{l(i)}}$ can be defined as a mode-k tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{|V_{i_1}| \cdots |V_{i_{l(i)}}|}$ with

$$\boldsymbol{\mathcal{X}}_{\nu_1,\ldots,\nu_{l(i)}} = \begin{cases} 1 & \text{if } (\varphi_1^{-1}(\nu_1), \ldots, \varphi_{l(i)}^{-1}(\nu_{l(i)})) \in X \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where $\varphi_i^{-1}(\nu_i)$ denotes the mapping φ_i that corresponds to the i-th relation type, and $V_{i_1} \times \ldots \times V_{i_{l(i)}}$ are the indexes of the entity types used in the relation i.

2.1 MetaGraph

Following the above approach for $k = 2$, we would be considering only binary relations, which correspond to edges in the graph representation of the social network. Thus the adjacency matrix for such a graph would be resembled within a collection of mode-2 tensors.

MetaGraph introduced by [16] is a relational hypergraph representing multi-dimensional data in a network of entities. A MetaGraph is defined as a graph $G = (V, E)$, where each vertex corresponds to a set of entities of the same type and each edge is defined as a *hyper-edge* connecting two or more vertices. By the use of hyper-edges, the MetaGraph captures multi-dimensional relations of the social network and therefore provides a framework to model n-ary relations.

Given the notion of relation types defined above, each relation type $R_i = V_{i_1} \times \ldots \times V_{i_{l(i)}}$ corresponds to a hyper-edge in the MetaGraph G. Each relation type $R_i = (v_{i_1}, \ldots, v_{i_l})$ observed within the social network is reflected in a hyper-edge of the MetaGraph. Given a fixed set of relation types $R_1, \ldots, R_n$, we can model the occurrence of relations of type R_i by defining a Tensor $\boldsymbol{\mathcal{X}}^{(i)}$ for each R_i as described in (1).

This approach results in a description of the social network by means of different relational aspects $R_1, \ldots, R_n$. Each type R_i of relations for which a tensor is defined, reflects a subset of all the relations of the network. Capturing the complete set of relations among all entities would obviously result in $|\mathcal{P}(V)| = 2^{|V|}$ different tensors.

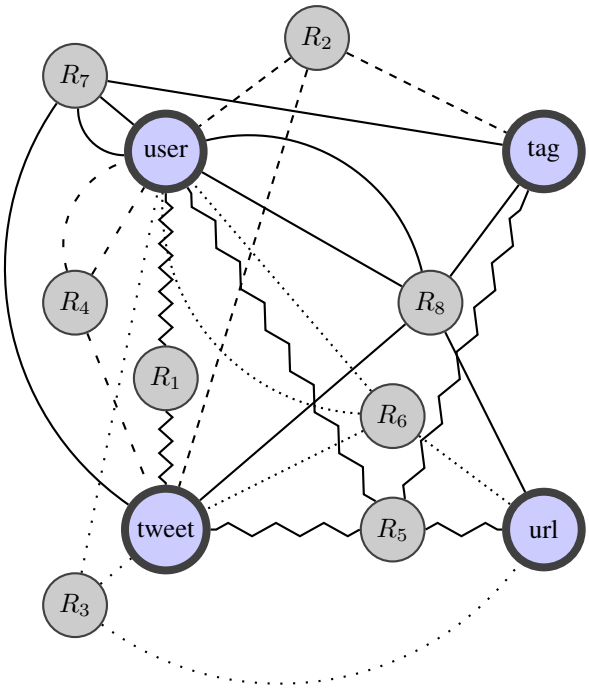


Figure 2: MetaGraph for *Twitter*

Figure 2 visualizes the metagraph we used for modeling possible relations in the microblogging framework *Twitter*. As an example, the relation R_8 referring to (*user*,*tweet*,*tag*,*url*) is represented by a hyper-edge connecting four vertices.

2.2 Community Discovery Problem

With the use of tensors we have an approximated description of our social networks by means of a set of relation types $R_1, \ldots, R_n$. Thus we can describe our network graph G by means of the data tensors which are defined according to the observed relations $R_1, \ldots, R_n$ in G, i.e.

$$G \mapsto \left\{ \boldsymbol{\mathcal{X}}^{(1)}, \ldots, \boldsymbol{\mathcal{X}}^{(n)} \right\}.$$

Based on this description we seek for a further partitioning of the tensor representation into clusters of entities.

The solution proposed in [16] is a factorization of the tensors $\boldsymbol{\mathcal{X}}^{(i)}$ into products of matrices $\boldsymbol{U}^{(q)}$ which share a global factor $[z]$ and some of the $\boldsymbol{U}^{(q)}$ matrices. Let $\boldsymbol{\mathcal{X}}^{(i)}$ be the tensor describing $V_{i_1} \times \ldots \times V_{i_{l(i)}}$, then we can factorize this as

$$\boldsymbol{\mathcal{X}}^{(i)} \approx [z] \prod_{j=1}^{l(i)} \times_j \boldsymbol{U}^{(i_j)}. \tag{2}$$

Within this factorization, the $[z]$ factor is a super-diagonal tensor containing non-zero values only at positions $(i, i, \ldots, i)$. The $\boldsymbol{U}^{(q)}$ are $\mathbb{R}^{|V_q| \times k}$ matrices, where $|V_q|$ is the number of entities of the q-th entity type and k is the number of communities we are looking for. For tensors which relate to relation types with overlapping entity types (e.g. (*user*,*keyword*,*tag*) and (*user*,*keyword*,*url*)) the corresponding factorizations share the related $\boldsymbol{U}^{(q)}$ matrices (e.g. $\boldsymbol{U}^{user}$ and $\boldsymbol{U}^{keyword}$). The $\times_j$ is the mode-j product of a tensor with a matrix.

With an appropriate normalization as used in [16], the $\boldsymbol{U}^{(q)}$ matrices only contain values of $[0, 1]$ which can be interpreted as probability values. Based on this, the value of $\boldsymbol{U}^{(q)}_{l,m}$ can be seen as the probability of entity $\varphi_q^{-1}(l)$ belonging to cluster $m \in \{1, \ldots, k\}$ and we can simply map an entity to its cluster $C(l)$ by

$$C(l) = \arg\max_m \boldsymbol{U}^{(q)}_{l,m}. \tag{3}$$

Thus, the community discovery is mapped onto the simultaneous factorization of a set of tensors. The objective is to find a factorized representation, which resembles the original data tensors $\{\boldsymbol{\mathcal{X}}^{(i)}\}$ as closely as possible. Given some distance measure $D : \mathbb{R}^{I_1 \times \ldots \times I_l} \times \mathbb{R}^{I_1 \times \ldots \times I_l} \to \mathbb{R}$ this leads to the following optimization problem:

$$\arg\min_{[z], \{\boldsymbol{U}^{(q)}\}} \sum_{i=1}^{n} D(\boldsymbol{\mathcal{X}}^{(i)}, [z] \prod_{j=1}^{l(i)} \times_j \boldsymbol{U}^{(i_j)}) \tag{4}$$

2.3 Batch Processing of Evolving Tensors

To capture the evolving behavior of the data tensors Lin et.al. proposed a batch processing approach. The stream is processed as disjoint sliding windows. Let t denote the current window and denote by $\mathbf{z}_{t-1}, \{\boldsymbol{U}^{(q)}_{t-1}\}$ the model obtained so far. Then the time-based optimization problem of [16] yielding the new model $\mathbf{z}_t, \{\boldsymbol{U}^{(q)}_t\}$ is given as

$$\arg\min_{\boldsymbol{z}_t, \{\boldsymbol{U}^{(q)}_t\}} (1-\alpha) \sum_{i=1}^{n} D(\boldsymbol{\mathcal{X}}^{(i)} [\boldsymbol{z}_t] \prod_{j=1}^{l(i)} \times_j \boldsymbol{U}^{(i)}_t) + \alpha L_{prior} \tag{5}$$

$$L_{prior} = D([\boldsymbol{z}_{t-1}][\boldsymbol{z}_t]) + \sum_{i=1}^{n} D(\boldsymbol{U}^{(i)}_{t-1} \boldsymbol{U}^{(i)}_t). \tag{6}$$

The L_{prior} reflects the divergence between the new and the old model, whereas the α specifies a trade-off between the models.

3 Tensors & Tensor Factorizations

The previous sections presented tensors as a mathematical structure to model multi-dimensional relations and motivated their use to describe multi-relational data such as community networks. In this section we will introduce tensors and their factorizations in more detail.

3.1 Tensor Decomposition

Tensors can be decomposed into a sum of component rank-one tensors [11]. A popular method for factorizing a tensor into a product of matrices is the PARAFAC decomposition (*CP*-decomposition) by [9]. Using CP, a third-order tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I \times J \times K}$ can be decomposed to

$$\boldsymbol{\mathcal{X}} = \left[\sum_{r=1}^{R} \boldsymbol{a}_r \circ \boldsymbol{b}_r \circ \boldsymbol{c}_r \right] + \boldsymbol{\mathcal{E}} \tag{7}$$

where R is a positive integer and $\boldsymbol{a}_r \in \mathbb{R}^I$, $\boldsymbol{b}_r \in \mathbb{R}^J$, $\boldsymbol{c}_r \in \mathbb{R}^K$. See Figure 3 for a schema of this decomposition. This decomposition is an approximation of $\boldsymbol{\mathcal{X}}$ by an error of $\boldsymbol{\mathcal{E}}$.

Neglecting this error tensor $\boldsymbol{\mathcal{E}}$ we are left with the approximation of $\boldsymbol{\mathcal{X}}$ by

$$\boldsymbol{\mathcal{X}} \approx \sum_{r=1}^{R} \boldsymbol{a}_r \circ \boldsymbol{b}_r \circ \boldsymbol{c}_r. \tag{8}$$

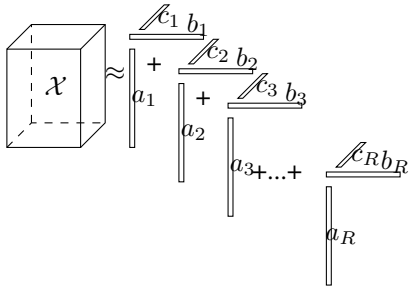


Figure 3: CP tensor decomposition

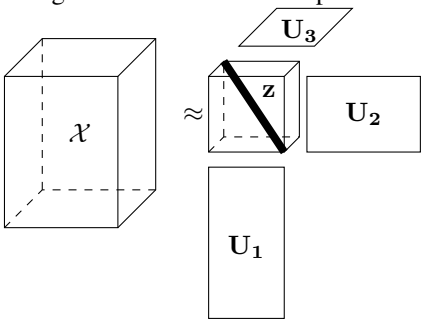


Figure 4: *CP* tensor decomposition incorporating weights

By breaking equation (8) further down into its elementwise form, we get

$$x_{ijk} \approx \sum_{r=1}^{R} a_{ir} b_{jr} c_{kr} \tag{9}$$

Since we later refer to the elements of this sum as probability values, we need to normalize the rank-one tensors $\boldsymbol{a}_r$, $\boldsymbol{b}_r$ and $\boldsymbol{c}_r$ for $r = 1, ..., R$ to length one. With this normalization, we are left with the following form

$$x_{ijk} \approx \sum_{r=1}^{R} z_r a_{ir} b_{jr} c_{kr} \tag{10}$$

where $\boldsymbol{z} \in \mathbb{R}^R$ is a weight-vector. This decomposition sometimes is called higher-order singular value decomposition (HOSVD) ([10]). The rank-one tensors $\boldsymbol{a}_r$, $\boldsymbol{b}_r$ and $\boldsymbol{c}_r$ for $r = 1, ..., R$ represent the singular values and can be used to derive a clustering of the data.

$\boldsymbol{a}_r$, $\boldsymbol{b}_r$ and $\boldsymbol{c}_r$ for $r = 1, ..., R$ can be combined in matrices $\boldsymbol{U}^1 \in \mathbb{R}^{I \times R}$, $\boldsymbol{U}^2 \in \mathbb{R}^{J \times R}$ and $\boldsymbol{U}^3 \in \mathbb{R}^{K \times R}$. We construct a super-diagonal tensor $[\boldsymbol{z}] \in \mathbb{R}^{R \times ... \times R}$ of $\boldsymbol{z}$ containing just zeros apart from positions $z_{k,...,k}$ where it contains value z_k. This allows us to write eq. (10) as *n-mode product* (see [11] for further information about tensor calculation):

$$\boldsymbol{\mathcal{X}} \approx ((([\boldsymbol{z}] \times_1 \boldsymbol{U}^1) \times_2 \boldsymbol{U}^2) \times_3 \boldsymbol{U}^3) = [\boldsymbol{z}] \prod_{i=1}^{3} \times_i \boldsymbol{U}^i \tag{11}$$

Figure 4 visualizes the *n-mode product* for better understanding.

3.2 METAFAC - Metagraph Factorization

As mentioned before, the Metagraph is a description of a multi-relational graph G by means of a set of tensors $\{\boldsymbol{\mathcal{X}}^{(i)}\}$. The objective of the METAFAC algorithm is to derive tensor decompositions of the $\boldsymbol{\mathcal{X}}^{(i)}$ with shared factors $[z], U^{(q)}$ which closely resemble the $\boldsymbol{\mathcal{X}}^{(i)}$. To measure the approximation, [16] proposed the Kullback Leibler divergence D_{KL} [13], thus implying the following optimization problem:

$$\underset{[z],\{U^{(q)}\}}{\arg\min} \sum_{i=1}^{n} D_{KL}(\boldsymbol{\mathcal{X}}^{(i)}, [\boldsymbol{z}] \prod_{i_1,...,i_{l(i)}} \times_j \boldsymbol{U}^{(i_j)}) \tag{12}$$

To solve for (12) the authors derived an approximation scheme by defining

$$\boldsymbol{\mu}^{(i)} = \text{vec}(\boldsymbol{\mathcal{X}}^{(i)} \oslash ([\boldsymbol{z}] \prod_{j=1}^{l(i)} \times_j \boldsymbol{U}^{(i_j)})) \tag{13}$$

$$\boldsymbol{\mathcal{S}}^{(i)} = \text{fold}(\boldsymbol{\mu}^{(i)} * (\boldsymbol{z} * \boldsymbol{U}^{M_i} * \cdots * \boldsymbol{U}^{1_i})^T) \tag{14}$$

where $\oslash$ is the elementwise division of tensors, and $*$ is the Khatri-Rao product of matrices. These values are then be used to update $\boldsymbol{z}$ and the $\{\boldsymbol{U}^{(q)}\}$ iteratively using

$$\boldsymbol{z} = \frac{1}{n} \sum_{i=1}^{n} \text{acc}(\boldsymbol{\mathcal{S}}^{(i)}, M_i + 1) \tag{15}$$

$$\boldsymbol{U}^q = \sum_{l:e_l \sim v_q} \text{acc}(\boldsymbol{\mathcal{S}}^{(i)}, q, M_e + 1) \tag{16}$$

where acc is the accumulation-function of tensors and $M_i + 1$ is the last mode of tensor $\boldsymbol{\mathcal{S}}^{(i)}$. This update is carried out iteratively until the the sum in 4 converges. The optimization is shown in Algorithm 1.

Algorithm 1 MF algorithm

Input: Meta-Graph $G = (V, E)$, data tensors $\{\boldsymbol{\mathcal{X}}^{(e)}\}$
Output: z and $\{\boldsymbol{U}^q\}$

procedure METAFAC(G, $\{\boldsymbol{\mathcal{X}}^{(e)}\}$)
  Initialize z, $\{\boldsymbol{U}^q\}$
  repeat
    for all $e \in E$ **do**
      compute $\{\boldsymbol{\mathcal{S}}^{(e)}\}$ by eq. (13), (14)
      compute z by eq. (15)
    end for
    for all $q \in V$ **do**
      update $\{\boldsymbol{U}^q\}$ by eq. (16)
    end for
  until *convergence*
end procedure

The batched version of the METAFAC approximation can be derived by using the KL-divergence in equations (5),(6). An appropriate approximation scheme has been proposed by the following update function:

$$\boldsymbol{z} = (1 - \alpha) \sum_{i=1}^{n} \text{acc}(\boldsymbol{\mathcal{S}}^{(i)}, M_i + 1) + \alpha \boldsymbol{z}_{t-1} \tag{17}$$

$$\boldsymbol{U}^{(q)} = (1 - \alpha) \sum_{l:e_l \sim v_q} \text{acc}(\boldsymbol{\mathcal{S}}^{(j)}, q, M_i + 1) + \alpha \boldsymbol{U}^{(q)}_{t-1} \tag{18}$$

4 Stream-based Community Discovery with Tensors

In this section we present our adaptions of the METAFAC framework by introducing a sampling-based tensor representation of graphs and using time-stamped relations to induce a decrease of impact of relations to reflect the decreasing importance of *Twitter* messages.

Given a social network we are provided with a sequence M of messages

$$M := \langle m_0, m_1, \ldots \rangle$$

where each message m_i implies a set of relations $\mathcal{R}(m_i)$. Let $\tau(m_i) \geq 0$ be the arrival time of m_i. This results in an overall sequence of relations

$$S := \langle \mathcal{R}(m_0), \mathcal{R}(m_1), \ldots \rangle$$

which are continuously added to the evolving social network graph G. Hence we are faced with a sequence

$$\langle G_{t_0}, G_{t_1}, \ldots \rangle$$

of graphs where each G_{t_i} contains the relations of all messages up to time t_i.

Let t, t' be points in time with $t < t'$. In the following we will by $G_{[t,t']}$ denote the graph implied by only the messages of time-span $[t, t']$, hence $G_t = G_{[0,t]}$. Accordingly the graphs are represented by the corresponding tensor as

$$G_{[t,t']} \mapsto \left\{ \boldsymbol{\mathcal{X}}^{(1)}, \ldots, \boldsymbol{\mathcal{X}}^{(n)} \right\}_{[t,t']}.$$

4.1 The MFSTREAM Algorithm

The METAFAC approach uses a sliding window of some fixed window size w_s to manage streams. Given a sequence of time-points t_j for $j \in \mathbb{N}$ with $t_j = t_{j-1} + w_s$, it factorizes $\{\boldsymbol{\mathcal{X}}^{(i)}\}_{[t_{j-1}, t_j]}$ based on a trade-off factor α as denoted in equation (5).

Our MFSTREAM algorithm interleaves the optimization of METAFAC by adding new relations during optimization and uses a time-based weighting function to take into account the relations' decreasing importance. Additionally, the optimization is carried out over only a partial set of relations as older relations tend to become obsolete for adjusting the model. We will present the time-based weighting and the sampling strategy in the following and present the complete algorithm in 4.4.

4.2 Time-based Relation Weighting

So far we considered the property of two or more entities to be related as binary property, i.e. if entities i, j and k are related, then

$$\boldsymbol{\mathcal{X}}_{i,j,k} = w,$$

with $w \in \{0, 1\}$. With the extraction of relations from time-stamped messages – as provided within the *Twitter* platform – we are interested in incorporating the age of these relations to reflect the decreasing up-to-dateness of the information.

Hence we associate each relation $r \in R_i$ with a timestamp $\tau(r)$ of the time at which this relation has been created (i.e. the time of the message from which it has been extracted). With S being a set of relations extracted from messages this leaves us with the tensor representation of relation type $R_i = V_{i_1} \times \ldots \times V_{i_{l(i)}}$ as

$$\boldsymbol{\mathcal{X}}^{(i)}_{i_1,\ldots,i_{l(i)}} = \begin{cases} \tau(r) & \text{if } r = (\varphi_1^{-1}(\nu_1), \ldots, \varphi_{l(i)}^{-1}(\nu_{l(i)})) \in S \\ 0 & \text{otherwise} \end{cases} \quad (19)$$

In addition to that, we introduce a *global clock*, denoted by $\tau_{\max}$, which represents the largest (i.e. the most recent) timestamp of all relations observed so far:

$$\tau_{\max} := \max \{ \tau(r) \mid r \in X \}.$$

Storing the timestamp $\tau(r)$ for each entry r in the tensors allows us to define a weighting function for the relations based on the global clock value. A simple example for a parametrized weighting function is given as

$$\omega_{\alpha,\beta}(r) := \frac{\alpha}{\alpha + \frac{1}{\beta}(\tau_{\max} - \tau(r))}. \quad (20)$$

4.3 Sampling

The runtime of each iteration of the approximation scheme is basically manifested by the maximum number N of non-zero entries in the tensors. To reduce the overall optimization time, we restrict the size of the tensors, i.e. number of entities of each type, by introducing constants $C_q \in \mathbb{N}$ and providing new entity mappings φ_q by

$$\varphi_q : V_q \to \overline{V}_q \text{ with } \overline{V}_q = \{1, \ldots, C_q\}.$$

This has two implications: Clearly, these φ_q will not be bijective anymore if $|V_q| > C_q$. Moreover, the size of the $\boldsymbol{U}^{(q)}$ matrices will also be limited to $C_q \times k$.

We deal with these imposed restrictions by defining dynamic entity mappings φ_q, which maps a new entity e (i.e. an entity that has not been mapped before) to the next free integer of $\{1, \ldots, C_q\}$. If no such element exists, we choose $f \in \{1, \ldots, C_q\}$ as the element that has longest been inactive, i.e. not been mapped to by φ_q. The relations affected by f are then removed from all tensors and the current cluster model, i.e. $\boldsymbol{U}^{(q)}_{f,i} = \frac{1}{k} \; \forall \; i = 1, \ldots, k$.

Effectively this introduces a "current window" $\{\boldsymbol{\mathcal{X}}^{(i)}\}$ of relations, that affect the adaption of the clustering in the next iteration. In contrast to the original METAFAC approach this also frees us from having to know the number of entities and a mapping of the entities beforehand.

4.4 Continuous Integration

With the prerequisites of section 4.2 and 4.3 we now present our stream adaption MFSTREAM as Algorithm 2. MFSTREAM is a purely dynamic approach of METAFAC which adds new relations to the data tensors $\{\boldsymbol{\mathcal{X}}^{(q)}\}$ and fits the model $[z], \{\boldsymbol{U}^{(q)}\}$ after a specified number of T messages. This differentiates our approach from METAFAC as the optimization is performed by running a single iteration of the optimization loop – with respect to the time-based weighting – after adding the relations of T messages to the tensors. The time complexity per iteration of the MFSTREAM algorithm is the same as for the METAFAC algorithms (see Section 3.2). Due to the fixed tensor dimensions, the maximum number of non-zero elements N is constant, which implies $O(1)$ runtime.

5 Evaluation

For the evaluation of our approach we extracted relations of the *Twitter* website. *Twitter* is a blogging platform giving users the opportunity to inform other users by very small snippets of text containing a maximum of 140 characters. In spite of such limitations *users* are not only posting messages – called *tweets* – but also enriching their *tweets* by *tags*, *urls* or *mentions*, which allows users to address other *users*. This brings up the entity types {*user*, *tweet*, *tag*, *url* }.

To discover clusters on the above mentioned entities present on the *Twitter* platform, we constructed a meta-graph for *Twitter* as shown in Figure 2. The entity types

Algorithm 2 The MFSTREAM algorithm.

Input: MetaGraph $G = (V, E)$, Stream $M = \langle m_i \rangle$, capacities C_q, number of clusters k, constant $T \in \mathbb{N}$
procedure MFSTREAM
  Initialize z, $\{\boldsymbol{U}^{(q)}\}$, $c := 0$
  while $M \neq \emptyset$ **do**
    $m := m_c$, $c := c + 1$ ▷ *Pick the next message from the stream*
    for all $(r_{j_1}, \ldots, r_{j_{l(j)}}) \in \mathcal{R}(m)$ **do**
      for all $p = 1, \ldots, l(j)$ **do**
        if $\varphi_p(r_{j_p}) = nil$ **then** ▷ *Replacement needed?*
          if $|\varphi_p| = C_q + 1$ **then**
            $f^* := \arg\min_{f \in \varphi_p} \tau(f)$
            $\boldsymbol{U}^{(p)}_{f^*,s} := \frac{1}{k} \; \forall \; s = 1, \ldots, k$
          else
            $f^* := \min_{f \in \{1,\ldots,C_p\}} \varphi_p^{-1}(f) = nil$ ▷ *Pick next unmapped f^**
          end if
          $\varphi_p(r_{j_p}) := f^*$, $\tau(f^*) := \tau(m)$
        end if
      end for
    end for
    $\nu_i := \varphi_p(r_{j_i})$ for $i = 1, \ldots, l(j)$
    $\boldsymbol{\mathcal{X}}^{(p)}_{\nu_1,\ldots,\nu_{l(j)}} := \tau(m)$ ▷ *Update corresponding tensor*
    if $c \equiv 0 \mod T$ **then** ▷ *Single opt.-iteration every T steps*
      for all $i \in \{1, \ldots, n\}$ **do**
        compute $\{\boldsymbol{\mathcal{S}}^{(i)}\}$ by eq. (14) and (13)
        update z by eq. (15)
      end for
      for all $j \in \{1, \ldots, q\}$ **do**
        update $\{\boldsymbol{U}^{(j)}\}$ by eq. (16)
      end for
    end if
  end while
end procedure

- R_1: a *user* writing a *tweet*.
- R_2: a *user* writing a *tweet* containing a special *tag*.
- R_3: a *user* writing a *tweet* containing a special *url*.
- R_4: a *user* mentioning another *user* in a written *tweet*.
- R_5: a *user* writes a *tweet* containing a *tag* and an *url*.
- R_6: a *user* writing a *tweet* containing an *url* and mentioning another *user*.
- R_7: a *user* writes a *tweet* containing a *tag* and a mentioned *user*.
- R_8: a *user* mentioning another *user* in a *tweet* containing a *tag* and an *url*.

themselves imply as much as $\mathcal{P}(V) = 2^4$ possible relation types, some of which will not arise or are redundant. E.g. since each *tweet* is written by a user, there is no relation (*tweet*,*tag*) which does not also refer to a *user*. Hence, our MetaGraph is based on the relation types $\{R_1, \ldots, R_8\}$ given as:

We extracted 1000 seed users and their direct *friends* and *followers*. *Followers* are following a *user* which means that messages of the *user* are directly visible for the *followers* at their *twitter* website. *Friends* are all the users a particular *user* is following. We used an English stopword filter to extract users which are writing in English language and processed all *friends* and *followers* of the seed users, revealing about 478.000 *users*. For these, we extracted all the messages written between the 19th and 23rd of February 2010. Out of these 2.274.000 *tweets* we used the *tweets* written at the 19th of February for our experiments, leaving about 389.000 *tweets* from 41.000 *users*.

5.1 Evaluating the Model

For a comparison of the clusterings produced by MFSTREAM and the METAFAC approaches we employ the "within cluster" point scatter [8]. This is given as

$$W(C) := \sum_{k=1}^{K} N_k \sum_{C(i)=k} \parallel x_i - \overline{x}_k \parallel^2 \qquad (21)$$

where K is the number of clusters, x_i is a member of a cluster $C(i)$ and $\overline{x}_k$ is the centroid of a cluster k. It can be seen as a sum of dissimilarities between elements in the particular clusters.

We created clusterings on a stream of 200k messages with MFSTREAM and restricted the tensor dimensions to $C_{user} = C_{tweet} = 5000$ and $C_{tag} = C_{url} = 1000$. We employed several weighting functions such as $\omega_{1,1}, \omega_{10,1000}$ and $\omega_{100,1000}$ as well as a binary weighting which equals the unweighted model (i.e. $w \in \{0, 1\}$).

To be able to compare the clusterings of MFSTREAM and METAFAC, we processed messages until the first entity type V_i reached its limit and stored the resulting clustering on disk. Then we reset the φ mappings and started anew, revealing a new clustering every time an entity type i reached C_i, revealing a total of 93 clusterings. We applied METAFAC on the messages that have been used to create these 93 clusterings and computed their similarities using $W(C)$. Figure 5 shows that MFSTREAM delivers results comparable to METAFAC for different weighting functions. Figure 7 shows that using timestamped values instead of binary values for calculation of the MFSTREAM delivers better results. The decrease of T, which implies a larger number of optimization steps, intuitively increases the quality

Weighting	$W(C)$ (mean)	std. deviation
METAFAC	$5.685 \cdot 10^7$	$1.32 \cdot 10^7$
binary	$7.511 \cdot 10^7$	$2.00 \cdot 10^7$
$\omega_{1,1}$	$6.142 \cdot 10^7$	$1.57 \cdot 10^7$
$\omega_{\mathbf{1,1000}}$	$\mathbf{6.002 \cdot 10^7}$	$\mathbf{1.38 \cdot 10^7}$
$\omega_{10,1000}$	$6.272 \cdot 10^7$	$1.43 \cdot 10^7$
$\omega_{100,1000}$	$6.724 \cdot 10^7$	$1.55 \cdot 10^7$

Figure 5: Mean $W(C)$ of different weights ($T = 20$), comparing MFSTREAM and METAFAC

T	$W(C)$ (mean)	std. deviation
5	$\mathbf{6.043 \cdot 10^7}$	$\mathbf{1.35 \cdot 10^7}$
10	$6.133 \cdot 10^7$	$1.36 \cdot 10^7$
50	$6.058 \cdot 10^7$	$1.40 \cdot 10^7$
100	$6.465 \cdot 10^7$	$1.49 \cdot 10^7$
250	$10.882 \cdot 10^7$	$3.13 \cdot 10^7$
500	$43.419 \cdot 10^7$	$11.47 \cdot 10^7$

Figure 6: Mean of $W(C)$ with different update steppings T (weight used: $\omega_{1,1000}$)

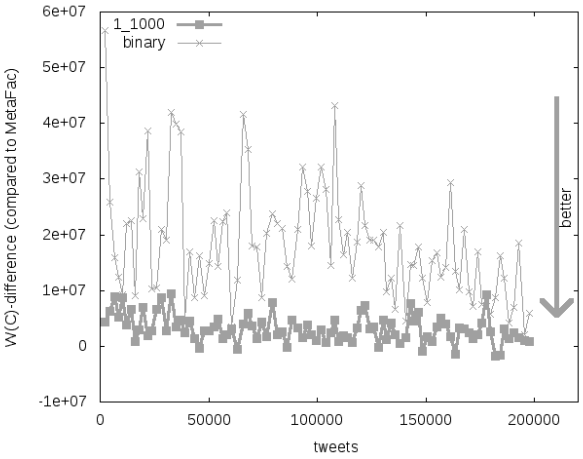


Figure 7: $W(C)$ for MFSTREAM compared to the METAFAC clusterings.

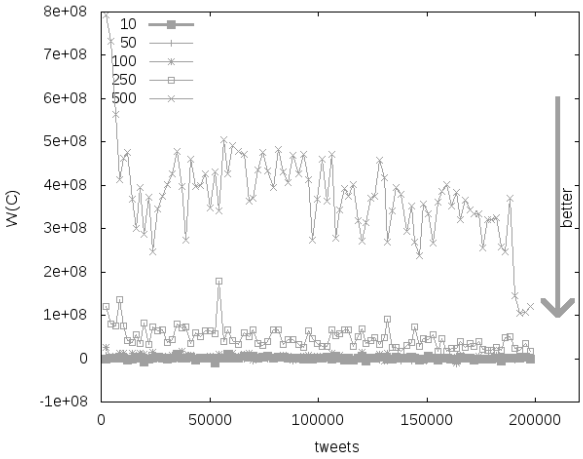


Figure 8: Relative $W(C)$ of MFSTREAM using different update step sizes T

of MFSTREAM as is attested by Figures 6 and 8.

In addition, we made experiments to show the effect of the update frequency T. Figure 9 shows the relative runtime of MFSTREAM where $T = 1$ corresponds to the baseline at 1.0. Raising T results in shorter runtime, since the model is updated less frequently, which is the major time factor. The upper curve shows the runtime for updating after 5 relations ($T = 5$), the middle one shows $T = 10$, and the latter refers to $T = 50$.

Varying sizes of entity types by the C_q results in clusterings of different numbers of entities, which cannot be directly compared by $W(C)$. Hence, we normalized $W(C)$ by the variance $\mathcal{V}$ of each clustering. Larger models of course incorporate more information, which results in more stable clusterings as can be seen in Figure 10.

6 Conclusion and Future Work

In this work we presented MFSTREAM, a flexible algorithm for clustering multi-relational data from evolving networks, derived from the METAFAC framework by [16]. The main improvement of our approach is the reduction of the approximation scheme on to a small relevant window of relations. The proposed time-based weighting of relations contributes to this reduction by removing obsolete information that is not relevant to the model adaption anymore.

MFSTREAM is able to handle relations containing new, unseen entities by offering a replacement strategy for the set of entities considered at optimization time. This makes it especially suitable to continuously integrate new data from a stream. We evaluated MFSTREAM on real-world data crawled from the *Twitter* platform and showed its comparability to METAFAC.

The use of backend storage for off-loading obsolete data that can be re-imported into the optimization window at a later stage might be an interesting advancement. Also, concurrent criteria *runtime* and *quality* offer a starting point for multi-objective optimization. Additionally, recent works [15] motivate further improvements to handle a dynamic number k of clusters within MFSTREAM.

References

[1] E. Acar, S. A. Çamtepe, M. S. Krishnamoorthy, and B. Yener. Modeling and multiway analysis of chatroom tensors. In *ISI*, pages 256–268, 2005.

[2] E. Acar, S. A. Çamtepe, and B. Yener. Collective sampling and analysis of high order tensors for chatroom communications. In *ISI*, pages 213–224, 2006.

[3] J. F. Allen. Towards a general theory of action and time. *Artificial Intelligence*, 23:123–154, 1984.

[4] B. Bader, M. W. Berry, and M. Browne. *Survey of Text Mining II*, chapter Discussion tracking in Enron email using PARAFAC, pages 147–163. Springer, 2007.

[5] A. Banerjee, S. Basu, and S. Merugu. Multi-way clustering on relation graphs. In *SDM*. SIAM, 2007.

[6] D. Cai, X. He, and J. Han. Tensor space model for document analysis. In E. N. Efthimiadis, S. T. Dumais, D. Hawking, and K. Järvelin, editors, *SIGIR*, pages 625–626. ACM, 2006.

[7] R. Harshman. Foundations of the parafac procedure: Models and conditions for an "explanatory" multi-

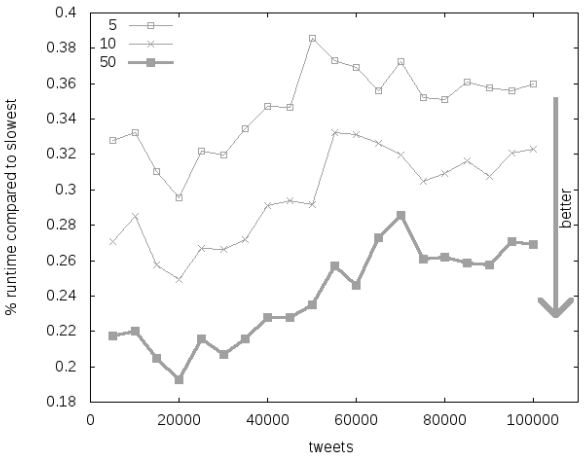


Figure 9: Relative runtime of MFSTREAM using different numbers of relations for update.

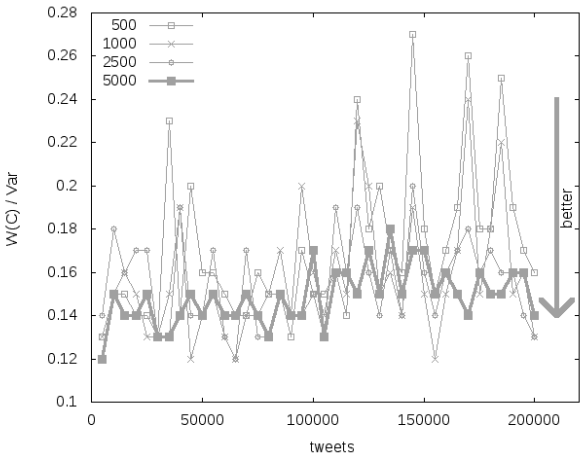


Figure 10: $W(C)/\mathcal{V}$ of MFSTREAM using different sizes of models.

modal factor analysis. *UCLA Working Papers in Phonetics*, 16, 1970.

[8] T. Hastie, R. Tibshirani, and F. J. The elements of statistical learning-data mining, inference and prediction. *Springer, Berlin Heidelberg New York*, 2001.

[9] H. Kiers. Towards a standardized notation and terminology in multiway analysis. *Journal of Chemometrics*, 14(3):105–122, 2000.

[10] E. Kilmer and C. D. Moravitz Martin. Decomposing a tensor. *SIAM News*, 37(9), 2004.

[11] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, September 2009.

[12] T. G. Kolda, B. W. Bader, and J. P. Kenny. Higher-order web link analysis using multilinear algebra. In *ICDM '05: Proceedings of the Fifth IEEE International Conference on Data Mining*, pages 242–249, 2005.

[13] S. Kullback and R. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22:79–86, 1951.

[14] L. D. Lathauwer, B. D. Moor, and J. Vandewalle. A multilinear singular value decomposition. *SIAM J. Matrix Anal. Appl.*, 21(4):1253–1278, 2000.

[15] Y.-R. Lin, J. Sun, N. Cao, and S. Liu. Contextour: Contextual contour visual analysis on dynamic multi-relational clustering. In *Proceedings of the SIAM Conference on Data Mining (SDM10)*, 2010. Not published, yet.

[16] Y.-R. Lin, J. Sun, P. Castro, R. Konuru, H. Sundaram, and A. Kelliher. Metafac: Community discovery via relational hypergraph factorization. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge discovery and data mining (SIGKDD 2009)*, pages 527–536, Paris, France, 2009. ACM.

[17] A. Shashua and T. Hazan. Non-negative tensor factorization with applications to statistics and computer vision. In *ICML '05: Proceedings of the 22nd international conference on Machine learning*, pages 792–799, New York, NY, USA, 2005. ACM.

[18] J. Sun, D. Tao, and C. Faloutsos. Beyond streams and graphs: dynamic tensor analysis. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 374–383, New York, NY, USA, 2006. ACM.

[19] X. Wang, J.-T. Sun, Z. Chen, and C. Zhai. Latent semantic analysis for multiple-type interrelated data objects. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 236–243, New York, NY, USA, 2006. ACM.

Listing closed sets of strongly accessible set systems with applications to data mining*

Mario Boley[1,2] **Tamás Horváth**[1,2] **Axel Poigné**[2] **Stefan Wrobel**[1,2]

[1]Dept. of Computer Science III, University of Bonn, Germany
[2]Fraunhofer IAIS, D-53754 Sankt Augustin, Germany
{mario.boley,tamas.horvath,axel.poigne,stefan.wrobel}@iais.fraunhofer.de

Abstract

We study the problem of listing all closed sets of a closure operator σ that is a partial function on the power set of some finite ground set E, i.e., $\sigma : \mathcal{F} \to \mathcal{F}$ with $\mathcal{F} \subseteq \mathcal{P}(E)$. A very simple divide-and-conquer algorithm is analyzed that correctly solves this problem if and only if the domain of the closure operator is a strongly accessible set system. Strong accessibility is a strict relaxation of greedoids as well as of independence systems. This algorithm turns out to have delay $O(|E|(T_{\mathcal{F}} + T_\sigma + |E|))$ and space $O(|E| + S_{\mathcal{F}} S_\sigma)$, where $T_{\mathcal{F}}$, $S_{\mathcal{F}}$, T_σ, and S_σ are the time and space complexities of checking membership in $\mathcal{F}$ and computing σ, respectively. In contrast, we show that the problem becomes intractable for accessible set systems. We relate our results to the data mining problem of listing all support-closed patterns of a dataset and show that there is a corresponding closure operator for all datasets if and only if the set system satisfies a certain confluence property. submissions.

*A long version of this extended abstract appeared in *Theoretical Computer Science* **411**(3), 691–700, 2010.

Mining Music Playlogs for Next Song Recommendations

Andre Busche, Artus Krohn-Grimberghe, Lars Schmidt-Thieme
University of Hildesheim, Germany
{busche, artus, schmidt-thieme}@ismll.uni-hildesheim.de

Abstract

Recommender systems are popular social web tools, as they address the information overload problem and provide personalization of results [1]. This paper presents a large-scale collaborative approach to the crucial part in the playlist recommendation process: next song recommendation. We show that a simple markov-chain based algorithm improves performance compared to baseline models when neither content, nor user metadata is available. The lack of content-based features makes this task particularly hard.

1 Introduction

Nowadays, mobile devices enable ubiquitous access to any piece of music any time and anywhere. Additionally, online stores offer millions of song tracks to those that want fresh content. But which of this plethora shall the store suggest or the user choose? How to cope with the information overload problem? Recommender systems come as a rescue: they aim at providing the user with relevant music content, based on personal listening preferences.

Given the usage pattern for mobile music devices, in the extremes two different scenarios emerge that music recommender systems should tackle: on the one hand, there is the interactive listening scenario, where users actively use the system and want to influence the next song being played; on the other hand, there is the passive listening scenario, where music is just an ambient factor and the users want to receive full playlists instead of single tracks.

In the interactive listening scenario, e.g., music online stores aim at increasing their revenue by recommending tracks to users who did not purchase these songs before. Another use case may be a music player automatically suggesting candidates for the next song to play.

In the passive listening scenario, portable devices are often used to listen to one's own music in a passive way while doing something else, e.g., sports. Collaborative listening portals like last.fm[1] or Pandora[2] automatically generate music playlists given some seed song or artist, and provide the user with a sequential but uncontrollable list of tracks to listen to. One of their aims is to get users in contact to novel tracks they did not know before, possibly with the aim to sell those.

'Automatic playlist generation' is involved in all of the cases above as a means of generating a probable sequence or set of songs to play (next) given some fixed but potentially huge set of songs in a library. While all these applications are of the same kind, the available data for them used to differ considerably: Online stores and listening portals usually had no information about the music library of the user and, traditionally, portable devices were only able to choose next tracks from the users' music library for playlist generation. The availability of nearly ubiquitous internet, though, changed this picture where services like iTunes or Zune in combination with the respective devices have access to both, the user's music library and the vast libraries of online stores.

In this paper, we exploit massive collaborative usage data gathered on portable devices to recommend songs a user might want to listen to next. In the recommendation lists, songs available in the user's library as well as songs not yet existing in his library are considered. Thus we allow for utility maximization of the user's library and for cross-selling opportunities for the music store the user is connected to.

We can show that collaborative recommendation techniques can greatly improve performance over baseline methods in situations where content features are unavailable. Furthermore, we demonstrate that for next song recommendation denser data yields better performance.

2 Related Work

Playlist generation has been studied with great interest for about the last five years by many researchers, each focusing on different recommendation techniques. A content-based approach to derive playlists by providing the system with one start and end song has been introduced in [3]. The authors strive at providing smooth transitions between genres: Having MFCC coefficients[4] from the start and end song at hand, they model the songs by a single Gaussian and compute an 'optimal transition playlist' by means of a (weighted) sum of 'optimal' divergence ratios. Figuratively, their approach recommends those songs which are closest to the λ-weighted 'connection line' between the songs. The evaluation in [3] is based on whether the generated playlists only contain songs from source genre, target genre, or both. However, it has been shown in [9] that while experts and social communities commonly agree when (manually) classifying songs between distinct genres at a coarse level (e.g. 'Classic', or 'Rock' songs), their agreement decreases when it comes to the classification of songs to subgenres.

An algorithm for learning possible song transitions from radio station playlists was introduced in [5]. While their approach is close to ours on a conceptual level, it is radically

[1] http://www.last.fm/

[2] http://www.pandora.com/

different w.r.t. the features used. They use content features such as the MFCC coefficients and others to model song transition probabilities based on the content, rather than collaborative play information and available metadata as in our approach. To this end, they are more concerned enabling their approach to also work with previously unknown content, for which no collaborative data yet exists. This problem, known as the 'new-item'-problem for recommender systems, as well as the 'new-user' problem [8], is not handeled here.

Additionally, [5] incorporate tag information when recommending playlist items. Tags have become a cornerstone of the Web 2.0 and, furthermore, they increase recommender performance for item prediction (e.g. next song prediction) [6]. A manual way to get 'gold-standard' tags is described in [10]. Regrettably, tags are not available in our dataset.

While there is much related work in the area of movie recommendation (e.g. [11]), we deem this task to be fundamentally different: One the one hand, short video clips and music tend to be consumed in a session-oriented way. On the other hand, TV programme length movies are mostly consumed one at a time.For the former scenario, it can be assumed that within a session a user's mood remains in more or less the same state. For the latter, the time spans are considered too large for this to be necessarily the case.

3 Methodology

In this section we describe our technique for next song recommendation based on playlogs. For training data we restrict our algorithms to collaborative data. We use the Zune dataset which was sponsored by Microsoft Zune[3]. It contains an anonymized sample of multimedia play log entries from Sep '08 to Feb '09 plus additional metadata on the multimedia items. With 1.3 billion log entries this dataset is quite large, leveraging information from 1.3 million users $u \in \mathcal{U}$ on 44 million song tracks $t \in \mathcal{T}$. Further does it contain playlists generated by external experts. These playlists comprise songs grouped by, e.g., themes. However, neither content-information nor tags are available in our sample.

For the remainder of this work, we only consider a subset of the data: First, we only use songs, ignoring the other multimedia content. Second, of the songs we restrict ourselves to those contained in the predefined playlists. Furthermore, we only consider users who have played at least 60 songs, possibly spread over multiple sessions $\mathcal{S}$. This filtering reduces the data to a 'dense' part, composed of approx. 500k users u and 26k tracks t. 18.4 million (1.4% of the original) log entries relate these users to items. This yields our largest training and sole evaluation dataset, 'Playlog'.

Our data also contains some hand-made playlists created by experts. Since the playlist information is explicitly given, we can directly transform them into sessions for later use. From the playlists, we derive two further datasets:

- 'Playlist (sequential)': The sequential occurrence within the playlist, as initially ordered by the expert.
- 'Playlist (combinatorial)': The set-based interpretation of the playlists.

We define the density of our data to be the fraction of present transitions to the amount of possible transitions

$$\text{density}(\mathcal{S}) := \frac{\mid \mathcal{S} \mid}{\mid \mathcal{T} \mid^2}$$

with $\mathcal{T}$ being the filtered set of tracks ($\mid \mathcal{T} \mid \sim 26k$). Let a 'transition' from song t_i to song t_{i+1} happen iff song t_i is played for its whole duration, and song t_{i+1} directly follows t_i, possibly with some lag (see below).

Table 1 gives an overview of the densities within our derived datasets.

Technique	Density
Playlog	2.2%
Playlists (sequential)	0.0001%
Playlists (combinatorial)	0.003%

Table 1: Varying Densities for the derived datasets

3.1 Reconstructing Session Data

We define a session (consecutive plays of tracks t by a user u) as $s_{u,m} = \{t_1, \ldots, t_{T_s}\} \in \mathcal{S}^u$ with m many sessions for user u. If the session $s_{u,m}$ is obvious from the context, we omit the subscripts for readability. The tracks in one session are ordered w.r.t. $t_i \prec t_{i+1} \equiv \text{abs}(\text{end}(t_i) - \text{start}(t_{i+1})) < g \quad \forall i = 1, \ldots, T_s - 1$, some $g \geq 0$ and start(·) and end(·) being the start and end timestamp from the logfile, respectivly.

No explicit session information is available in the data. To derive session data, we empirically choose some $g > 0$ based on the lag distribution in the log file.

The lag between two songs is assumed to be (close to) 0, if two songs are enqueued in a playlist and played after the other. This, however, is not always true, as depicted in Figure 1.

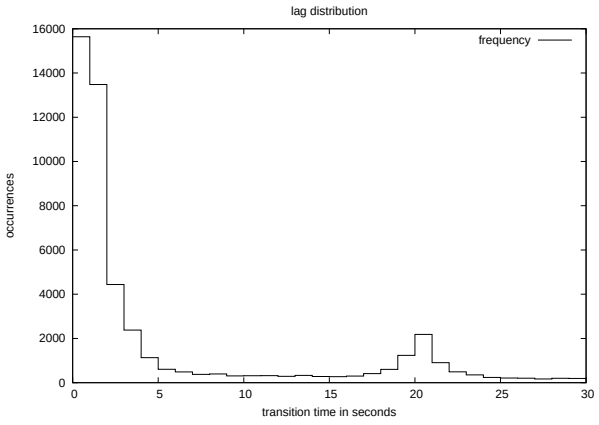


Figure 1: Gap between consecutive songs

Two effects can easily be seen in the chart: First, there are several lags close to 0 seconds. Second, there is a small increase around the lag of 20 seconds. We interpret lags close to 0 to come from the data collection mechanism, that is, log entries are rounded to the nearest second, and the devices need some time to (pre)load the next song to play, or cross-fades are used. Gaps around the duration of 20 seconds are assumed to define 'end-of-playlist'-markers, that is, the last song within a playlist ends, the user recognises 'silence' after a short amount of time, locates his device, and loads/starts another playlist. Since there is no indication that a consecutive playlist is thematically close to the

[3] http://www.zune.net

one before, we assume $g = 10$ seconds for session splitting.

Doing so we are following standard literature: E.g., in [5], track transitions are mined from online radio stations (possibly also containing advertisements). They use lags of 20 seconds. In [7], it is assumed that a session breaks if the elapsed time between two songs exceeds 5 minutes.

Using this technique, we yield 8.8 million sessions from the 18.4 million log entries. As it will become clear in Section 4, we only use those sessions having at least two tracks. We also require each user to contribute at least two sessions. This way we can avoid the 'new user' problem.

Figure 2 shows the occurrences of different session lengths in our data (See the 'all-pair' line there).

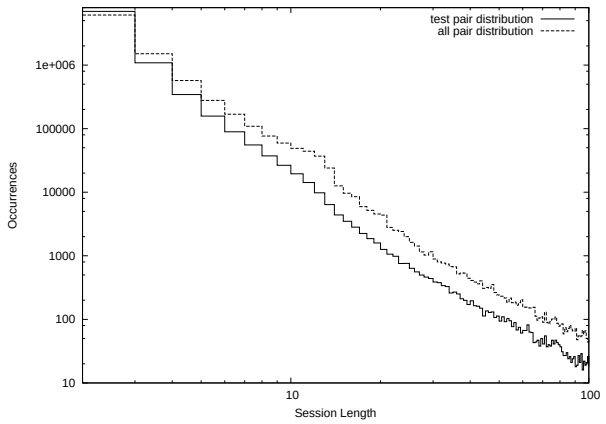


Figure 2: Distribution of session lengths

3.2 Baselines

For comparison with the later presented algorithms, we use the following 'most popular' (MP) baseline methods for next song prediction:

- most often downloaded/purchased songs,
- most often played songs
- most popular by user

Our 'most popular by user' baseline predicts those tracks to the user that he has listened to most. The baseline methods are only trained on the playlog, do not consider sequence information available in the logfile and always output a constant prediction.

3.3 Markov-based prediction

We use a Markov-Chain model of length 1 to incorporate sequence information in the prediction task. We define the probability of the occurrence of the song t_{i+1}, given track t_i, as

$$\hat{p}(t_{i+1}|t_i) = \frac{\mid \{(t'_i, t'_{i+1}) \in \mathcal{S} | t'_i = t_i, t'_{i+1} = t_{i+1}\} \mid}{\mid \{(t'_i, t'_{i+1}) \in \mathcal{S} | t'_i = t_i\} \mid}$$

with $\mathcal{S} = \cup_{u \in \mathcal{U}} \mathcal{S}^u$ when the algorithm is trained on 'Playlog' and When trained on the two playlists variants, $\mathcal{S}$ becomes the union of all transitions derived from the playlists, in either the sequential, or the combinatorial way as described in section 3.

Taking Top-N predictions into account, we define our classifier as

$$\hat{t}_{i+1}(t_i, N) := \arg\max^N_{t \in \mathcal{T}} \hat{p}(t_{i+1} = t | t_i)$$

with a slight abuse of notation as we let arg max return a (ranked) list of the N most probable tracks. Ties in $\hat{p}(t_{i+1}|t_i)$ are broken arbitrarily.

4 Evaluation

4.1 Evaluation protocol

We splitted our data using the 'leave-one-session-out' paradigm. The main idea is to predict either the next or all following songs, given some start/seed songs in that session. In an application scenario, predicting the next song means that we are able to reconstruct the listening history of the user under changing conditions (the recommendation is adjusted each time a next song is played). By predicting multiple following songs, we aim at measuring whether we are able to capture the 'mood' or 'style' of the users' listening preferences for the current session.

In both protocols, we average results over each session $s_m \in \mathcal{S}$ (each having T_s many tracks) by choosing some 'test song' t_j with $j+1 \in [1, T_s - 1]$. Training is done using all other sessions $\mathcal{S} \setminus \{s_m\}$, and songs $\{t_1, \ldots, t_j\} \in s_m$.

Using this technique yielded 4.5 million training sequences (consecutive song plays) and 8.8 million test sequences. Please note that we have more test sequences than training sequences. This necessarily is the case, since we required j to be in $[1, T_s - 1]$, and thus, considering a session of length 2 (our minimal required length), we always use song t_1 as a seed for the recommendation algorithm to predict song t_2.

Figure 2 shows that the test set distribution ('test-pair': the distribution of $j + 1$ plotted on x-axis) reflects the true session length distribution of the session data extracted from the log file. The reason for having more test sequences *of length 2* than actual sessions of length 2 is that each session of length greater than 2 can possibly be reduced to create a test pair for $j = 1$.

Predicting the next song

The first evaluation focuses on next-song prediction, i.e. only song t_{j+1} should be predicted by our algorithm. To evaluate the performance, we calculate the precision of the first N recommended tracks

$$P@N := \text{avg}_{s_m \in \mathcal{S}} I(t^{s_m}_{j+1} \in \hat{t}_{j+1}(t_j, N))$$

with $I(\cdot)$ being 1 if the argument is true, otherwise 0. The precision calculates whether one of the n recommended tracks matches the true following track.

Top-N List intersection

When evaluating against all following songs of a session, we use Top-N List intersection as evaluation measure. It is able to capture whether our recommendation algorithm correctly identifies the 'mood' of the user. It evaluates whether $\hat{t}_{j+1}(t_j)$ matches for some $o \geq j + 1$. Thus, the measure can also be thought of whether our algorithms are able to identify songs a user likes, or dislikes, given the current session/'mood'. Its definition measures the overlap of the recommendations with all following songs within a session and is given by

$$list(N) := \sum_{s_m \in \mathcal{S}} \frac{\mid \hat{t}_{j+1}(t_j, N) \cap \{t_o \in s_m : o \geq j + 1\} \mid}{min(\mid \hat{t}_{j+1}(t_j, N) \mid, T_s - j, N)},$$

again with $N = 5$.

Please note that both lists in the measure may contain less than 5 tracks. Possible reasons are recommendation of less than 5 tracks ($|\hat{t}_{j+1}(t_j, N)| < 5$), or too few tracks to the end of the session ($T_s - j < 5$) in the test data.

4.2 Results

Results for our experiments are given in Table 2.

Algorithm	P@5	list(5)
Markov$_{global}$	43.9	61.8
Markov$_{playlists_sequential}$	11.5	17.6
Markov$_{playlists_combinatorial}$	14.0	23.9
MP$_{user}$	3.2	7.9
MP$_{download}$	1.2	1.8
MP$_{playcount}$	1.2	1.8

Table 2: Results for P@5 and list(5) measure in %

A closer look at the predictions for MP$_{download}$ and MP$_{playcount}$ revealed that the recommended tracks are identical. We consider our filtering process as described in Section 3 as being too restrictive. Furthermore, note that the most popular by user results are not further ranked: A user having heard each of his tracks only once introduces ties which can not be broken in a smart fashion.

What is more interesting is a comparison between our Markov-based recommendations. For next song prediction, the global variant taking collaborative information into account clearly exceeds both variants being trained on the predefined playlists. This is supported from the density estimations given in Table 1, since playlists contain much less sequential information than the log files.

The overall increase of performance for the combinatorial playlist variant over the sequential one for both evaluation measures show that playlists within our data are used to group tracks, rather to reflect sequential transitions.

Having a closer look at the performance of the Markov$_{global}$ recommender broken down by session lengths gives interesting insights to the data, as shown in Figure 3.

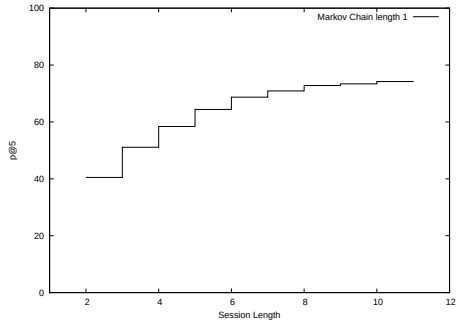


Figure 3: Results for different session length (j) for Markov$_{global}$.

As expected from a collaborative model, performance increases when the available historic data in the sessions also increases. While we have shown in Figure 2 that the total amount of test data decreases for increasing j, the overall trend is still surprising, since we have not considered Markov-lengths > 1 here. We assume that the derived sessions being longer than 2 are based on random plays of only a few tracks, i.e. repetitive sequences are present among different sessions.

5 Conclusion and Future Work

In this work we have presented our initial algorithm to recommend songs to the user to listen to next. We showed that simple collaborative information greatly improves performance compared to baseline methods.

As it has been shown in literature [2][5], incorporating actual content data might help in the recommendation task. However, up to our best knowledge, there is no clear and obvious way on how to do so.

Furthermore, we currently limited our research to only use implicit metadata derived from listening history data, rather than using, e.g., user-defined tags to guide the recommendation process. Further gathering of metadata, be it either content or tag data, will result in the need to define (multiple) similarity measures, which need to be combined in some smart way to improve the recommendation process. Initial studies in this area have been done, e.g., in [5]. However, while their approach shows that learning and using such similarity measures can help, they also implicitly show that current content-based solutions have lots of space for further improvements. To this end, we aim at adding more metadata to our system.

References

[1] Adomavicius G., Tuzhilin, A.: Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions, IEEE Transactions on Knowledge and Data Engineering, Vol 17, No 6, June 2005 pp. 734-749

[2] Casey, M. A., et al.: Content-Based Music Information Retrieval: Current directions and Future Challenges, Proceedings of the IEEE, Vol 96, No. 4, April 2008, pp. 668-696.

[3] Flexer, A., et al.: Playlist Generation Using Start and End Songs, ISMIR 2008, pp. 173-178.

[4] Logan, B.: Mel Frequency Cepstral Coefficients for Music Modelling, ISMIR 2000.

[5] Maillet, F., et al.: Steerable Playlist Generation by Learning Song Similarity from Radio Station Playlists, ISMIR 2009, pp. 345-350.

[6] Nanopoulos A., Krohn-Grimberghe A.: Recommending in Social Tagging Systems based on Kernelized Multiway Analysis, IFCS 2009.

[7] Ragno, R., Burges, CJC, Herley, C.: Inferring similarity between music objects with application to playlist generation, ACM MIR 2005, pp. 73-80.

[8] Schein et al.: Methods and metrics for cold-start recommendations, SIGIR 2002, pp. 253-260.

[9] Sordo et al.: The Quest for Musical Genres: Do the Experts and the Wisdom of the Crowds Agree?, ISMIR 2008, pp. 255-260.

[10] Turnbull, D., Barrington, L., Lanckriet, G.: Five Approaches to Collecting Tags for Music, ISMIR 2008, pp. 225-230.

[11] Yang, B., et al.: Online video recommendation based on multimodal fusion and relevance feedback, ACM CIVR 2007, pp. 73-w80.

Graded Multilabel Classification: The Ordinal Case*

Weiwei Cheng, Krzysztof Dembczynski, and Eyke Hüllermeier
Marburg University
Hans-Meerwein-Str., 35032 Marburg, Germany
{cheng, dembczynski, eyke }@informatik.uni-marburg.de

Abstract

We propose a generalization of multilabel classification that we refer to as *graded multilabel classification*. The key idea is that, instead of requesting a yes-no answer to the question of class membership or, say, relevance of a class label for an instance, we allow for a *graded membership* of an instance, measured on an ordinal scale of membership degrees. This extension is motivated by practical applications in which a graded or partial class membership is natural. Apart from introducing the basic setting, we propose two general strategies for reducing graded multilabel problems to conventional (multilabel) classification problems. Moreover, we address the question of how to extend performance metrics commonly used in multilabel classification to the graded setting, and present first experimental results.

1 Introduction

Problems of *multilabel classification* (MLC), in which an instance may belong to several classes simultaneously or, say, in which more than one label can be attached to a single instance, are ubiquitous in everyday life: At IMDb, a movie can be categorized as *action*, *crime*, and *thriller*, a CNN news report can be tagged as *people* and *political* at the same time, etc. Correspondingly, MLC has received increasing attention in machine learning in recent years.

In this paper, we propose a generalization of MLC that we shall refer to as *graded multilabel classification* (GMLC). The key idea is that, instead of requesting a yes-no answer to the question of class membership or, say, relevance of a class label for an instance, we allow an instance to belong to a class *to a certain degree*. In other words, we allow for graded class membership in the sense of *fuzzy set theory* [Zadeh, 1965]. In fact, there are many applications for which this extension seems to make perfect sense. In the case of movie genres, for example, it is not always easy to say whether or not a movie belongs to the category *action*, and there are definitely examples which can be considered as "almost action" or "somewhat action". Another obvious example comes from one of the benchmark data sets in MLC, namely the *emotions* data [Trohidis *et al.*, 2008]. Here, the problem is to label a song according to the Tellegen-Watson-Clark model of mood: amazed-surprised, happy-pleased, relaxing-clam, quiet-still, sad-lonely, and angry-aggressive.

It is important to emphasize that the relevance of a label is indeed *gradual* in the sense of fuzzy logic and not *uncertain* in the sense of probability theory. The latter would mean that, e.g., a song is either relaxing or it is not—one is only uncertain about which of these two exclusive alternatives is correct. As opposed to this, gradualness is caused by the vagueness of categories like "relaxing song" and "action movie", and means that one does not have to fully agree on one of the alternatives. Instead, one can say that a song is somewhere in-between (and can be certain about this).

As will be explained in more detail later on, our idea is to replace simple "yes" or "no" labels by membership degrees taken from a finite ordered scale such as

$$M = \{\texttt{not at all}, \texttt{somewhat}, \texttt{almost}, \texttt{fully}\}. \quad (1)$$

Admittedly, graded multilabel data sets of that kind are not yet widely available. We believe, however, that this is a kind of hen and egg problem: As long as there are no methods for learning from graded multilabel data, new data sets will be created in the common way, possibly forcing people to give a "yes" or "no" answer even when they are hesitating.

The rest of this paper is organized as follows: The problem of multilabel classification is introduced in a more formal way in Section 2. In Section 3, we propose our graded generalization of MLC and, moreover, outline two different strategies for reducing GMLC problems to conventional (multilabel) classification problems. In Section 4, we address the question of how to extend MLC evaluation metrics from the conventional to the graded setting. Finally, Section 5 presents some first experimental results.

2 Multilabel Classification

Let $\mathbb{X}$ denote an instance space and let $\mathcal{L} = \{\lambda_1, \lambda_2, \ldots, \lambda_m\}$ be a finite set of class labels. Moreover, suppose that each instance $\boldsymbol{x} \in \mathbb{X}$ can be associated with a subset of labels $L \in 2^{\mathcal{L}}$; this subset is often called the set of *relevant* labels, while the complement $\mathcal{L} \setminus L$ is considered as *irrelevant* for $\boldsymbol{x}$. Given training data in the form of a finite set T of observations in the form of tuples $(\boldsymbol{x}, L_{\boldsymbol{x}}) \in \mathbb{X} \times 2^{\mathcal{L}}$, typically assumed to be drawn independently from an (unknown) probability distribution on $\mathbb{X} \times 2^{\mathcal{L}}$, the goal in multilabel classification is to learn a classifier $H : \mathbb{X} \longrightarrow 2^{\mathcal{L}}$ that generalizes well beyond these observations in the sense of minimizing the expected prediction loss with respect to a specific loss function; examples of commonly used loss functions include the *subset*

*This paper has been presented at International Conference on Machine Learning, Haifa, Israel, 2010.

zero-one loss, which is 0 if $H(\boldsymbol{x}) = L_{\boldsymbol{x}}$ and 1 otherwise, and the *Hamming loss* that computes the percentage of labels whose relevance is incorrectly predicted:

$$\mathrm{E}_H(H(\boldsymbol{x}), L_{\boldsymbol{x}}) = \frac{1}{|\mathcal{L}|}\big|H(\boldsymbol{x}) \,\Delta\, L_{\boldsymbol{x}}\big|, \tag{2}$$

where Δ is the symmetric difference between sets.

An MLC problem can be reduced to a conventional classification problem in a straightforward way, namely by considering each label subset $L \in 2^{\mathcal{L}}$ as a distinct (meta-)class. This approach is referred to as *label powerset* (LP) in the literature. An obvious drawback of this approach is the potentially large number of classes that one has to deal with in the newly generated problem; obviously, this number is $2^{|\mathcal{L}|}$ (or $2^{|\mathcal{L}|} - 1$ if the empty set is excluded as a prediction). This is the reason why LP typically works well if the original label set $\mathcal{L}$ is small but quickly deteriorates for larger label sets [Tsoumakas and Vlahavas, 2007].

Another way of reducing multilabel to conventional classification is offered by the *binary relevance* (BR) approach. Here, a separate binary classifier H_i is trained for each label $\lambda_i \in \mathcal{L}$, reducing the supervision to information about the presence or absence of this label while ignoring the other ones. For a query instance $\boldsymbol{x}$, this classifier is supposed to predict whether λ_i is relevant for $\boldsymbol{x}$ ($H_i(\boldsymbol{x}) = 1$) or not ($H_i(\boldsymbol{x}) = 0$). A multilabel prediction for $\boldsymbol{x}$ is then given by $H(\boldsymbol{x}) = \{\lambda_i \in \mathcal{L} \,|\, H_i(\boldsymbol{x}) = 1\}$. Since binary relevance learning treats every label independently of all other labels, an obvious disadvantage of this approach is its ignorance of correlations and interdependencies between labels.

Many approaches to MLC learn a multilabel classifier H in an indirect way via a scoring function $f : \mathbb{X} \times \mathcal{L} \longrightarrow \mathbb{R}$ that assigns a real number to each instance/label combination. The idea is that a score $f(\boldsymbol{x}, \lambda)$ is in direct correspondence with the probability that λ is relevant for $\boldsymbol{x}$. Given a scoring function of this type, multilabel prediction can be realized via thresholding:

$$H(\boldsymbol{x}) \;=\; \{\lambda \in \mathcal{L} \,|\, f(\boldsymbol{x}, \lambda) \geq t\} \;\;,$$

where $t \in \mathbb{R}$ is a threshold. As a byproduct, a scoring function offers the possibility to produce a ranking (weak order) $\succeq_{\boldsymbol{x}}$ of the class labels, simply by sorting them according to their score:

$$\lambda_i \succeq_{\boldsymbol{x}} \lambda_j \;\Leftrightarrow\; f(\boldsymbol{x}, \lambda_i) \geq f(\boldsymbol{x}, \lambda_j) \;\;. \tag{3}$$

Sometimes, this ranking is even more desirable as a prediction, and indeed, there are several evaluation metrics that compare a true label subset with a predicted ranking instead of a predicted label subset; an example is the *rank loss* which computes the average fraction of label pairs that are not correctly ordered:

$$\mathrm{E}_R(f, L_{\boldsymbol{x}}) = \frac{\sum_{(\lambda,\lambda') \in L_{\boldsymbol{x}} \times \overline{L}_{\boldsymbol{x}}} S(f(\boldsymbol{x}, \lambda), f(\boldsymbol{x}, \lambda'))}{|L_{\boldsymbol{x}}| \times |\overline{L}_{\boldsymbol{x}}|},$$

where $\overline{L}_{\boldsymbol{x}} = \mathcal{L} \setminus L_{\boldsymbol{x}}$ is the set of irrelevant labels and $S(u, v) = 1$ if $u < v$, $= 1/2$ if $u = v$, and $= 0$ if $u > v$. The idea to solve both problems simultaneously, ranking and MLC, has recently been addressed in [Fürnkranz *et al.*, 2008]: A *calibrated ranking* is a ranking with a "zero point" separating a positive (relevant) part from a negative (irrelevant) one.

3 Graded Multilabel Classification

Generalizing the above setting of multilabel classification, we now assume that each instance $\boldsymbol{x} \in \mathbb{X}$ can belong to each class $\lambda \in \mathcal{L}$ *to a certain degree*. In other words, the set $L_{\boldsymbol{x}}$ of relevant labels is now a fuzzy subset of $\mathcal{L}$. This fuzzy set is characterized by a membership function, namely an $\mathcal{L} \longrightarrow M$ mapping, where M is the set of graded membership degrees. For notational simplicity, we shall not distinguish between the fuzzy set $L_{\boldsymbol{x}}$ and its membership function, and denote by $L_{\boldsymbol{x}}(\lambda)$ the degree of membership of the label $\lambda \in \mathcal{L}$ in the fuzzy set $L_{\boldsymbol{x}}$.

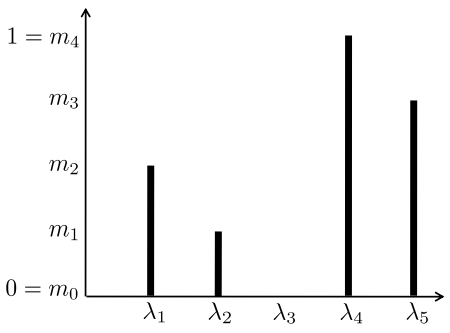


Figure 1: Vertical reduction, viz. prediction of membership degree (ordinate) for each label (abscissa).

In fuzzy set theory, the set of membership degrees is supposed to form a complete lattice and is normally taken as the unit interval (i.e., $M = [0, 1]$ endowed with the standard order). Here, however, we prefer an *ordinal scale* of membership degrees, that is, a finite ordered set of membership degrees such as (1). More generally, we assume that $M = \{m_0, m_1, \ldots, m_k\}$, where $m_0 < m_1 < \ldots < m_k$ (and $m_0 = 0$ and $m_k = 1$ have the special meaning of zero and full membership). In the context of multilabel classification, an ordinal membership scale is arguably more convenient from a practical point of view, especially with regard to data acquisition. In fact, people often prefer to give ratings on an ordinal scale like (1) instead of choosing precise numbers on a cardinal scale.

The goal, now, is to learn a mapping $H : \mathbb{X} \longrightarrow \mathcal{F}(\mathcal{L})$, where $\mathcal{F}(\mathcal{L})$ is the class of fuzzy subsets of $\mathcal{L}$ (with membership degrees in M). Following the general idea of *reduction* [Balcan *et al.*, 2008], we seek to make GMLC problems amenable to conventional multilabel methods via suitable transformations. There are two more or less obvious possibilities to reduce graded multilabel classification to conventional (multilabel) classification. In agreement with the distinction between the "vertical" description of a fuzzy subset F of a set U (through the membership function, i.e., by specifying the degree of membership $F(u)$ for each element $u \in U$) and the "horizontal" description (via level cuts $[F]_\alpha = \{u \in U \,|\, F(u) \geq \alpha\}$), we distinguish between a vertical and a horizontal reduction.

3.1 Vertical Reduction

Recall the *binary relevance* approach to conventional MLC: For each label $\lambda_i \in \mathcal{L}$, a separate binary classifier H_i is trained to predict whether this label is relevant ($H_i(\boldsymbol{x}) = 1$) or not ($H_i(\boldsymbol{x}) = 0$) for a query instance $\boldsymbol{x} \in \mathbb{X}$. Generalizing this approach to GMLC, the idea is to induce a classifier

$$H_i : \mathbb{X} \longrightarrow M \tag{4}$$

for each label λ_i. For each query instance $\boldsymbol{x} \in \mathbb{X}$, this classifier is supposed to predict the degree of membership of λ_i in the fuzzy set of labels $L_{\boldsymbol{x}}$. Instead of a binary classification problem, as in MLC, each classifier H_i is now

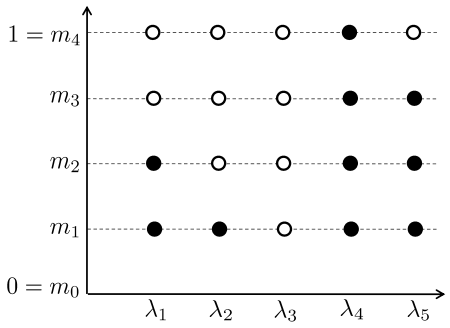


Figure 2: Horizontal reduction, viz. prediction of a subset of labels (indicated by black circles) on each level.

solving a multi-class problem. Since the target space M has an ordinal structure, these problems are *ordinal classification problems*. In other words, the vertical reduction of a GMLC problem eventually leads to solving a set of m (non-independent) ordinal classification problems; see Fig. 1 for an illustration.

Just like simple binary problems, ordinal classification problems are often solved indirectly via "scoring plus thresholding": First, a scoring function $f(\cdot)$ is learned, and k thresholds $t_1, \ldots, t_k$ are determined; then, for an instance $\boldsymbol{x}$, the i-th class is predicted if $f(\boldsymbol{x}, \lambda_i)$ is between t_{i-1} and t_i. Of course, if classifiers (4) are learned in this way, i.e., by inducing a scoring function $f(\cdot, \lambda_i)$ for each label λ_i, then these scoring functions can also be used to predict a ranking (3).

3.2 Horizontal Reduction

From fuzzy set theory, it is well-known that a fuzzy set F can be represented "horizontally" in terms of its level-cuts. This representation suggests another decomposition of a GMLC problem: For each level $\alpha \in \{m_1, m_2, \ldots, m_k\}$, learn the mapping

$$H^{(\alpha)} : \mathbb{X} \longrightarrow 2^M, \boldsymbol{x} \mapsto [L_{\boldsymbol{x}}]_\alpha \ . \tag{5}$$

Obviously, each of these problems is a *standard* MLC problem, since the level-cuts $[L_{\boldsymbol{x}}]_\alpha$ are standard subsets of the label set $\mathcal{L}$. Thus, the horizontal reduction comes down to solving k standard MLC problems; see Fig. 2 for an illustration.

It is worth mentioning that this decomposition comes with a special challenge. In fact, since level-cuts are nested in the sense that $[F]_\alpha \subset [F]_\beta$ for $\beta < \alpha$, the k MLC problems are not independent of each other. Instead, the predictions should be *monotone* in the sense that

$$(H^{(m_j)}(\boldsymbol{x}) = 1) \Rightarrow (H^{(m_{j-1})}(\boldsymbol{x}) = 1) \tag{6}$$

for all $j \in \{2, \ldots, k\}$. Thus, whenever a label λ_i is predicted to be in the m_j-cut of the fuzzy label set $L_{\boldsymbol{x}}$ associated with $\boldsymbol{x}$, it must also be in all lower level-cuts. Satisfying this requirement is a non-trivial problem. In particular, (6) will normally not be guaranteed when solving the k problems independently of each other.

Once an ensemble of k multilabel classifiers $H^{(m_1)}, \ldots, H^{(m_k)}$ has been trained, predictions can be obtained as follows:

$$H(\boldsymbol{x})(\lambda) \ = \ \max\{\, m_i \in M \mid \lambda \in H^{(m_i)}(\boldsymbol{x}) \,\} \tag{7}$$

Thus, the degree of membership of a label $\lambda \in \mathcal{L}$ in the predicted fuzzy set of labels associated with $\boldsymbol{x}$ is given by the maximum degree $m_i \in M$ for which λ is still in the predicted m_i-cut of this set.

The prediction of a ranking (3) is arguably less obvious in the case of the horizontal decomposition. Suppose that $f^{(m_1)}, \ldots, f^{(m_k)}$ are scoring functions trained on the k level cuts, using a conventional MLC method. As a counterpart to the monotonicity condition (6), we should require

$$f^{(m_1)}(\boldsymbol{x}, \lambda) \geq f^{(m_2)}(\boldsymbol{x}, \lambda) \geq \ldots \geq f^{(m_k)}(\boldsymbol{x}, \lambda) \tag{8}$$

for all $\boldsymbol{x} \in \mathbb{X}$ and $\lambda \in \mathcal{L}$. In fact, interpreting $f^{(m_i)}(\boldsymbol{x}, \lambda)$ as a measure of how likely λ is a relevant label on level m_i, this condition follows naturally from $[L_{\boldsymbol{x}}]_{m_1} \supset [L_{\boldsymbol{x}}]_{m_2} \supset \ldots \supset [L_{\boldsymbol{x}}]_{m_k}$. On each level m_i, the function $f^{(m_i)}(\boldsymbol{x}, \cdot)$ induces a ranking $\succeq_{\boldsymbol{x}}^{(m_i)}$ via (3), however, the identity $\succeq_{\boldsymbol{x}}^{(m_i)} \equiv \succeq_{\boldsymbol{x}}^{(m_j)}$ is of course not guaranteed; that is, $\succeq_{\boldsymbol{x}}^{(m_i)}$ may differ from $\succeq_{\boldsymbol{x}}^{(m_j)}$ for $1 \leq i \neq j \leq k$.

To obtain a global ranking, the level-wise rankings $\succeq_{\boldsymbol{x}}^{(m_i)}$ need to be aggregated into a single one. To this end, we propose to score a label λ by

$$f(\boldsymbol{x}, \lambda) \ = \ \sum_{i=1}^{k} f^{(m_i)}(\boldsymbol{x}, \lambda) \ . \tag{9}$$

This aggregation is especially reasonable if the scores $f^{(m_i)}(\boldsymbol{x}, \lambda)$ can be interpreted as probabilities of relevance $\mathbf{P}(\lambda \in [L_{\boldsymbol{x}}]_{m_i})$. Then, $f(\boldsymbol{x}, \lambda)$ simply corresponds to the *expected level* of $\boldsymbol{x}$, since

$$\sum_{i=1}^{k} f^{(m_i)}(\boldsymbol{x}, \lambda) = \sum_{i=1}^{k} \mathbf{P}(\lambda \in [L_{\boldsymbol{x}}]_{m_i}) =$$

$$= \sum_{i=1}^{k} \mathbf{P}(L_{\boldsymbol{x}}(\lambda) \geq m_i) = \sum_{i=1}^{k} i \cdot \mathbf{P}(L_{\boldsymbol{x}}(\lambda) = m_i)$$

Note, however, that we simply equated the levels m_i with the numbers i in this derivation, i.e., the ordinal scale $\mathcal{L}$ was implicitly embedded in a numerical scale by the mapping $m_i \mapsto i$ (on $\mathcal{L}$ itself, an averaging operation of this kind is not even defined). Despite being critical from a theoretical point of view, this embedding is often used in ordinal classification, for example when computing the absolute error $\mathrm{AE}(m_i, m_j) = |i - j|$ as a loss function [Lin and Li, 2007]. Interestingly, the absolute error is minimized (in expectation) by the *median* and, moreover, this estimation is invariant toward rescaling [Berger, 1985]. Thus, it does actually not depend on the concrete embedding chosen. Seen from this point of view, the median appears to be a theoretically more solid score than the mean value (9). However, it produces many ties, which is disadvantageous from a ranking point of view. This problem is avoided by (9), which can be seen an approximation of the median that breaks ties in a reasonable way.

3.3 Combination of Both Reductions

As mentioned above, the binary relevance approach is a standard (meta-)technique for solving MLC problems. Consequently, it can also be applied to each problem (5) produced by the horizontal reduction. Since BR can again be seen as a "vertical" decomposition of a regular MLC problem, one thus obtains a combination of horizontal and vertical decomposition: first horizontal, then vertical.

Likewise, the two types of reduction can be combined the other way around, first vertical and then horizontal. This is done by solving the ordinal classification problems

produced by the vertical reduction by means of a "horizontal" decomposition, namely a meta-technique that has been proposed by [Frank and Hall, 2001]: Given an ordered set of class labels $M = \{m_0, m_1, \ldots, m_k\}$, the idea is to train k binary classifiers. The i-th classifier considers the instances with label $m_0, \ldots, m_{i-1}$ as positive and those with label $m_i, \ldots, m_k$ as negative.

Interestingly, both combinations eventually coincide in the sense of ending up with the same binary classification problems. Roughly speaking, a single binary problem is solved for each label/level combination $(\lambda_i, m_j) \in \mathcal{L} \times M$ (each circle in the picture in Fig. 2), namely the problem to decide whether $L_{\boldsymbol{x}}(\lambda_i) \leq m_j$ or $L_{\boldsymbol{x}}(\lambda_i) > m_j$. Any difference between the two approaches is then due to different ways of aggregating the predictions of the binary classifiers. In principle, however, such differences can only occur in the case of inconsistencies, i.e., if the monotonicity condition (6) is violated.

3.4 Generalizing IBLR-ML

Our discussion so far has been restricted to meta-techniques for reducing GMLC to MLC problems, without looking at concrete methods. Nevertheless, there are several methods that can be generalized immediately from the binary to the gradual case. As an example, we mention the IBLR-ML method that will also be used in our experiments later on. This method, which was recently proposed in [Cheng and Hüllermeier, 2009], combines instance-based learning with logistic regression and again trains one classifier H_i for each label. For the i-th label λ_i, this classifier is derived from the logistic regression equation

$$\log\left(\frac{\pi_0^{(i)}}{1-\pi_0^{(i)}}\right) \;=\; \omega_0^{(i)} + \sum_{j=1}^{m} \gamma_j^{(i)} \cdot \omega_{+j}^{(i)}(\boldsymbol{x}_0) \;\;, \quad (10)$$

where $\pi_0^{(i)}$ denotes the (posterior) probability that λ_i is relevant for $\boldsymbol{x}_0$, and

$$\omega_{+j}^{(i)}(\boldsymbol{x}_0) = \sum_{\boldsymbol{x} \in \mathcal{N}(\boldsymbol{x}_0)} \kappa(\boldsymbol{x}_0, \boldsymbol{x}) \cdot y_j(\boldsymbol{x}) \quad (11)$$

is a summary of the presence of the j-th label λ_j in the neighborhood of $\boldsymbol{x}_0$; here, κ is a kernel function, such as the (data-dependent) "KNN kernel" $\kappa(\boldsymbol{x}_0, \boldsymbol{x}_i) = 1$ if $\boldsymbol{x}_i \in \mathcal{N}_k(\boldsymbol{x}_0)$ and $= 0$ otherwise, where $\mathcal{N}_k(\boldsymbol{x}_0)$ is the set of k nearest neighbors of $\boldsymbol{x}_0$. Moreover, $y_j(\boldsymbol{x}) = +1$ if λ_j is present (relevant) for the neighbor $\boldsymbol{x}$, and $y_j(\boldsymbol{x}) = -1$ in case it is absent (non-relevant). Obviously, this approach is able to capture interdependencies between class labels: The estimated coefficient $\gamma_j^{(i)}$ indicates to what extent the relevance of label λ_i is influenced by the relevance of λ_j. A value $\gamma_j^{(i)} > 0$ means that the presence of λ_j makes the relevance of λ_i more likely, i.e., there is a positive correlation. Correspondingly, a negative coefficient would indicate a negative correlation. Given a query instance $\boldsymbol{x}_0$, a multilabel prediction is made on the basis of the predicted posterior probabilities of relevance: $H(\boldsymbol{x}_0) = \{\lambda_i \in \mathcal{L} \,|\, \pi_0^{(i)} > 1/2\}$.

This approach can be generalized to the GMLC setting using both the horizontal and the vertical reduction. The vertical reduction leads to solving an ordinal instead of a binary logistic regression problem for each label, while the horizontal reduction comes down to solving the following k multilabel problems ($r = 1, \ldots, k$):

$$\log\left(\frac{\pi_0^{(i,r)}}{1-\pi_0^{(i,r)}}\right) \;=\; \omega_0^{(i,r)} + \sum_{j=1}^{m} \gamma_j^{(i,r)} \omega_{+j}^{(i,r)}(\boldsymbol{x}_0) \quad (12)$$

Recall, however, that these problems are not independent of each other. Solving them simultaneously so as to guarantee the monotonicity constraint (6) is an interesting but non-trivial task. In the experiments in Section 5, we therefore derived independent predictions and simply combined them by (7).

4 Loss Functions

As mentioned before, a number of different loss functions have already been proposed within the setting of MLC. In principle, all these functions can be generalized so as to make them applicable to the setting of GMLC. In this section, we propose extensions of some important and frequently used measures. Moreover, we address the question of how to handle these extensions in the context of the horizontal and vertical reduction technique, respectively.

4.1 Representation of Generalized Losses

To generalize the Hamming loss (2), it is necessary to replace the symmetric difference operator defined on sets, Δ, by the symmetric difference between two fuzzy sets. This can be done, for example, by averaging over the symmetric differences of the corresponding level-cuts, which in our case leads to

$$\mathrm{E}_H^*(H(\boldsymbol{x}), L_{\boldsymbol{x}}) = \frac{\sum_{i=1}^{k} \left|[H(\boldsymbol{x})]_{m_i} \,\Delta\, [L_{\boldsymbol{x}}]_{m_i}\right|}{k|\mathcal{L}|}. \quad (13)$$

Note that this "horizontal" computation can be replaced by an equivalent "vertical" one, namely

$$\mathrm{E}_H^*(H(\boldsymbol{x}), L_{\boldsymbol{x}}) = \frac{\sum_{i=1}^{|\mathcal{L}|} \mathrm{AE}(H(\boldsymbol{x})(\lambda_i), L_{\boldsymbol{x}}(\lambda_i))}{k|\mathcal{L}|}, \quad (14)$$

where $\mathrm{AE}(\cdot)$ is the absolute error of a predicted membership degree which, as mentioned above, is defined by $\mathrm{AE}(m_i, m_j) = |i - j|$. In other words, minimizing the symmetric difference level-wise is equivalent to minimizing the absolute error label-wise.

It is worth to mention that the existence of an equivalent horizontal and vertical representation of a loss function, like in the case of (13) and (14), is not self-evident. For example, replacing in (14) the absolute error on the ordinal scale M by the simple 0/1 loss leads to

$$\mathrm{E}_{0/1}^*(H(\boldsymbol{x}), L_{\boldsymbol{x}}) = \frac{1}{|\mathcal{L}|} \sum_{i=1}^{|\mathcal{L}|} \begin{cases} 0 & H(\boldsymbol{x})(\lambda_i) = L_{\boldsymbol{x}}(\lambda_i) \\ 1 & H(\boldsymbol{x})(\lambda_i) \neq L_{\boldsymbol{x}}(\lambda_i) \end{cases} .$$

Just like (14), this is a typical vertical expression of a loss function, that is, an expression of the form

$$\mathrm{A}\left(\{\ell\left(H(\boldsymbol{x})(\lambda_i), L_{\boldsymbol{x}}(\lambda_i)\right)\}_{i=1}^{|\mathcal{L}|}\right),$$

where $\ell(\cdot)$ is a loss defined on $\mathcal{L}$ and A is an aggregation operator. Interestingly, $\mathrm{E}_{0/1}^*$ does not have an equivalent horizontal representation. Thus, there is provably no loss function $L(\cdot)$ on $2^{\mathcal{L}}$ (and aggregation A) such that

$$\mathrm{E}_{0/1}^*(H(\boldsymbol{x}), L_{\boldsymbol{x}}) = \mathrm{A}\left(\{L\left([H(\boldsymbol{x})]_{m_i}, [L_{\boldsymbol{x}}]_{m_i}\right)\}_{i=1}^{k}\right).$$

This observation has an important implication. Namely, if the loss function to be minimized has a vertical but not a

horizontal representation, then a vertical decomposition of the learning problem is arguably more self-evident than a horizontal one, and vice versa. Strictly speaking, the non-existence of an equivalent representation does of course not exclude the existence of another loss function and aggregation operator producing the same predictions. Such alternatives, however, will normally be less obvious.

As an example of a loss function that lends itself to a horizontal representation, consider a variant of the Hamming loss based on the well-known Jaccard-index:

$$\mathrm{E}_J(H(\boldsymbol{x}), L_{\boldsymbol{x}}) = \frac{|H(\boldsymbol{x}) \cap L_{\boldsymbol{x}}|}{|H(\boldsymbol{x}) \cup L_{\boldsymbol{x}}|} \tag{15}$$

This variant avoids a certain disadvantage of the Hamming loss, which treats relevant and non-relevant labels in a symmetric way even though the former are typically less numerous than the latter, thereby producing a bias toward the prediction of non-relevance. A natural generalization of this measure is obtained by averaging (15) over the levels:

$$\mathrm{E}^*_J(H(\boldsymbol{x}), L_{\boldsymbol{x}}) = \frac{1}{k} \sum_{i=1}^{k} \frac{\left|[H(\boldsymbol{x})]_{m_i} \cap [L_{\boldsymbol{x}}]_{m_i}\right|}{\left|[H(\boldsymbol{x})]_{m_i} \cup [L_{\boldsymbol{x}}]_{m_i}\right|} \tag{16}$$

This extension, however, does not admit an equivalent vertical representation, which is plausible since the Jaccard-index is indeed a genuine set measure.

4.2 Rank Loss

The rank loss E_R can be generalized in a canonical way by the so-called C-index, which is commonly used as a measure of concordance in statistics [Gnen and Heller, 2005], and which is essentially equivalent to the pairwise ranking error introduced in [Herbrich *et al.*, 2000]:

$$\mathrm{E}^*_R(f, L_{\boldsymbol{x}}) = \frac{\sum_{i<j} \sum_{(\lambda,\lambda') \in M_i \times M_j} S(f(\boldsymbol{x}, \lambda), f(\boldsymbol{x}, \lambda'))}{\sum_{i<j} |M_i| \times |M_j|},$$

where $M_i = \{\lambda \in \mathcal{L} \,|\, L_{\boldsymbol{x}}(\lambda) = m_i\}$. As can be seen, the C-index is the fraction of labels that are correctly ordered by $f(\cdot)$: If label λ' has a higher degree of membership in $L_{\boldsymbol{x}}$ than λ, then the former should be ranked above the latter. It is also worth mentioning that the C-index has recently been proposed as a performance measure in the problem of *multipartite ranking* [Fürnkranz *et al.*, 2009], and indeed, the problem here can be considered as a problem of that kind when interpreting $\{M_0, M_1, \ldots, M_k\}$ as an ordered partition of the label set $\mathcal{L}$.

Other ranking losses proposed in the literature can be generalized, too. For example, the *one error* checks whether the top-ranked label is relevant or not:

$$\mathrm{E}_{1E}(f, L_{\boldsymbol{x}}) = \begin{cases} 0 & \arg\max_{\lambda \in \mathcal{L}} f(\boldsymbol{x}, \lambda) \in L_{\boldsymbol{x}} \\ 1 & \text{otherwise} \end{cases}$$

A natural generalization of this measure is obtained on the basis of the degree of membership of the top-ranked label in $L_{\boldsymbol{x}}$:

$$\mathrm{E}^*_{1E}(f, L_{\boldsymbol{x}}) = 1 - L_{\boldsymbol{x}}\left(\arg\max_{\lambda \in \mathcal{L}} f(\boldsymbol{x}, \lambda)\right) \quad .$$

5 Experimental Study

An experimental validation of the methods proposed in this paper is not at all straightforward. First, since we introduced a new machine learning problem, no benchmark data sets can be found so far. Essentially for the same reason, there are no existing methods to be used for comparison. The two reduction schemes proposed in Section 3, vertical and horizontal, are not easily comparable either, since these are meta-techniques using different types of base learners.

For these reasons, we decided to focus on another aspect, namely the general usefulness of the extended setting that we proposed in this paper. More specifically, our idea is to provide empirical evidence for the claim that allowing a user to label instances on a graded scale does provide useful extra information. In a sense, this claim is trivial if a prediction on a graded scale is eventually needed. For example, a reviewer recommendation (which can be seen as an estimation of the quality of a paper) on an ordered scale with labels such as "weak accept" and "strong accept" is normally more useful than just a "yes" or "no" answer to the question of acceptance.

However, we claim that *training* a learner on graded data can be useful even if only a *binary prediction* is eventually requested. Intuitively, this claim derives from the simple observation that graded data provides more information than binary data, which can be helpful, e.g., to determine proper decision boundaries.

5.1 Data

In light of the aforementioned lack of benchmark data, we used a data set from another research field, namely social psychology [Abele and Stief, 2004].[1] This data set, called BeLa-E, consists of 1930 instances and 50 attributes. Each instance corresponds to a graduate student. The first attribute is the sex of the student and the second one the age. Each of the other 48 attributes is a graded degree of importance of different properties of the future job, evaluated by the student on an ordinal scale with 5 levels ranging from 1 (completely unimportant) to 5 (very important). Examples of such properties include "reputation", "safety", "high income" and "friendly colleagues". Thus, every student was asked how important he or she considers these properties to be, and the student answered by assigning one of the aforementioned 5 levels.

On the basis of this data set, we generated (graded) MLC problems as follows: m of the above 48 attributes were randomly selected as the set of class labels, while all remaining $m - 48$ attributes plus the student's sex and age were taken as predictive features. The goal, then, is to train an MLC model that takes the features as input and produces a prediction of the relevance of the class labels as output.

Moreover, for every GMLC problem thus obtained, a binary version is produced by mimicking a student who is forced to answer either yes or no: The graded levels 1 and 2 are mapped to "No", the levels 4 and 5 are mapped to "Yes", and a coin is flipped for level 3.

5.2 Methods

As multilabel classifiers we used the IBLR-ML method outlined in Section 3.4 and, moreover, binary relevance learning with 10-nearest neighbor classification (BR-10NN) as base learner. Two types of learning are distinguished, binary and graded: In binary learning, the original data is first binarized as explained above (turning graded into 0/1 answers). Then, the multilabel classifier is trained on this data and used to make binary multilabel predictions. In graded learning, a GMLC classifier is trained

[1] The data set is available online at `http://www.uni-marburg.de/fb12/kebi/research`.

on the original (graded) data, using the horizontal reduction technique (for the BR learners automatically combined with the vertical reduction). The graded relevance predictions of these learners are then mapped to binary relevance degrees at the very end, using the same $M \longrightarrow \{0, 1\}$ mapping (randomized for label 3) as used in binary learning at the beginning. Eventually, both types of learning thus produce binary relevance predictions and, therefore, can be compared with each other.

5.3 Results

Each method was evaluated on a single problem in terms of a 10-fold cross validation. These evaluations were then averaged over a total number of 50 randomly generated problems. While averaging the performance over different data sets is questionable in general, we consider it legitimate in our case. In fact, all data sets are actually variants of the same problem, and indeed, the standard deviation of the performance was rather small throughout.

Table 1 summarizes the performance of the different methods for $m = 5$ and $m = 10$ in terms of the Hamming loss, subset zero-one loss, rank loss and C-index as performance metrics. As can be seen, the use of graded training data improves performance throughout, regardless of the learning method and the loss function. Comparing the respective mean values in terms of a paired t-test, the differences are significant at a significance level of 5%.

Note that, as an extension of the rank loss, the C-index is actually not intended for binary learning. We still included it, as it only requires a predicted ranking and a ground-truth labeling as input; thus, it can also be derived for the binary learner. Of course, this learner is at a disadvantage here, and indeed, the gains of the gradual learner for the C-index are slightly higher than those for the rank loss.

6 Summary and Conclusions

In this paper, we have proposed an extension of conventional multilabel classification, called *graded multilabel classification* (GMLC). The basic idea of GMLC is that the membership of an instance in a class or, say, the relevance of a label for an instance, is not a matter of "yes" or "no". Instead, the membership is measured on a *graded scale*, thus allowing for intermediate degrees of relevance. Here, we have focused on an *ordinal scale* as a special case, though numeric scales could in principle be used as well. In any case, a generalization of this kind appears to be useful and reasonable from a practical point of view.

Moreover, we have introduced two meta-techniques for reducing GMLC problems to existing machine learning problems, namely a vertical and a horizontal decomposition scheme. Whereas the former turns a GMLC problem into a set of ordinal classification problems, one for each label, the latter leads to solving a set of conventional multilabel problems, one for each level of the ordinal scale. In the context of these two techniques, we have also discussed the extension of MLC loss functions to the graded case.

Experimentally, we have shown that graded relevance does provide useful extra information from a learning point of view, even if only a binary prediction is requested. Collecting real-world GMLC data and complementing this study by further experiments is planned as future work. Besides, the GMLC framework gives rise to a number of interesting theoretical challenges, including but not limited to the simultaneous, monotonicity-preserving solution of the sub-problems produced by our reduction schemes.

Acknowledgments

We are grateful to Professor Abele-Brehm, University of Erlangen, for providing us the BELA-E data. This work has been supported by the Germany Research Foundation (DFG).

References

[Abele and Stief, 2004] A.E. Abele and M. Stief. Die Prognose des Berufserfolgs von Hochschulabsolventinnen und -absolventen. Befunde zur ersten und zweiten Erhebung der Erlanger Längsschnittstudie BELA-E. *Zeitschrift für Arbeits- und Organisationspsychologie*, 48:4–16, 2004.

[Balcan *et al.*, 2008] M.F. Balcan, N. Bansal, A. Beygelzimer, D. Coppersmith, J. Langford, and G.B. Sorkin. Robust reductions from ranking to classification. *Machine Learning*, 72(1–2):139–153, 2008.

[Berger, 1985] J.O. Berger. *Statistical Decision Theory and Bayesian Analysis*. Springer-Verlag, 2. edition, 1985.

[Cheng and Hüllermeier, 2009] W. Cheng and E. Hüllermeier. Combining instance-based learning and logistic regression for multilabel classification. *Machine Learning*, 76(2–3):211–225, 2009.

[Frank and Hall, 2001] E. Frank and M. Hall. A simple approach to ordinal classification. In *Proc. ECML–2001*, pages 145–156, Freiburg, Germany, 2001.

[Fürnkranz *et al.*, 2008] J. Fürnkranz, E. Hüllermeier, E. Mencia, and K. Brinker. Multilabel classification via calibrated label ranking. *Machine Learning*, 73(2):133–153, 2008.

[Fürnkranz *et al.*, 2009] J. Fürnkranz, E. Hüllermeier, and S. Vanderlooy. Binary decomposition methods for multipartite ranking. In *Proc. ECML/PKDD–2009*, Bled, Slovenia, 2009.

[Gnen and Heller, 2005] Mithat Gnen and Glenn Heller. Concordance probability and discriminatory power in proportional hazards regression. *Biometrika*, 92(4):965–970, 2005.

[Herbrich *et al.*, 2000] Ralf Herbrich, Thore Graepel, and Klaus Obermayer. Large margin rank boundaries for ordinal regression. In *Advances in Large Margin Classifiers*, pages 115–132. MIT Press, 2000.

[Lin and Li, 2007] H.T. Lin and L. Li. Ordinal regression by extended binary classiffications. In *Proc. NIPS–07*, pages 865–872, 2007.

[Trohidis *et al.*, 2008] K. Trohidis, G. Tsoumakas, G. Kalliris, and I. Vlahavas. Multilabel classification of music into emotions. In *Proc. Int. Conf. Music Information Retrieval*, 2008.

[Tsoumakas and Vlahavas, 2007] G. Tsoumakas and I. Vlahavas. Random k-labelsets: An ensemble method for multilabel classification. In *Proc. ECML–2007*, pages 406–417, Warsaw, 2007.

[Zadeh, 1965] L.A. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338–353, 1965.

Table 1: Performance (mean and standard deviation) in the case of $m = 5$ labels (above) and $m = 10$ labels (below).

	IBLR-ML		BR-10NN	
	binary	graded	binary	graded
Hamming loss	0.245±0.048	0.219±0.042	0.220±0.051	0.213±0.052
rank loss	0.190±0.062	0.180±0.057	0.328±0.115	0.310±0.104
C-index	0.204±0.047	0.183±0.045	0.381±0.089	0.361±0.080
subset zero-one loss	0.736±0.093	0.695±0.078	0.857±0.051	0.808±0.070
Hamming loss	0.225±0.017	0.207±0.018	0.230±0.018	0.217±0.018
rank loss	0.169±0.029	0.157±0.021	0.225±0.040	0.154±0.020
C-index	0.190±0.012	0.178±0.019	0.237±0.011	0.171±0.016
subset zero-one loss	0.908±0.028	0.875±0.042	0.913±0.022	0.893±0.034

Bootstrapping Noun Groups Using Closed-Class Elements Only

Kathrin Eichler
DFKI - Language Technology
Berlin, Germany
kathrin.eichler@dfki.de

Günter Neumann
DFKI - Language Technology
Saarbrücken, Germany
neumann@dfki.de

Abstract

The identification of noun groups in text is a well researched task and serves as a pre-step for other natural language processing tasks, such as the extraction of keyphrases or technical terms. We present a first version of a noun group chunker that, given an unannotated text corpus, adapts itself to the domain at hand in an unsupervised way. Our approach is inspired by findings from cognitive linguistics, in particular the division of language into open-class elements and closed-class elements. Our system extracts noun groups using lists of closed-class elements and one linguistically inspired seed extraction rule for each open class. Supplied with raw text, the system creates an initial validation set for each open class based on the seed rules and applies a bootstrapping procedure to mutually expand the set of extraction rules and the validation sets. Possibly domain-dependent information about open-class elements, as for example provided by a part-of speech lexicon, is not used by the system in order to ensure the domain-independency of the approach. Instead, the system adapts itself automatically to the domain of the input text by bootstrapping domain-specific validation lists. An evaluation of our system on the Wall Street Journal training corpus used for the CONLL 2000 shared task on chunking shows that our bootstrapping approach can be successfully applied to the task of noun group chunking.

1 Introduction

The identification of noun groups (or chunks) in text is a well researched task and serves as a pre-step for other natural language processing tasks, e.g. the extraction of keyphrases or technical terms as for example in [Eichler and Neumann, 2010]. Our approach to identifying noun groups in text is inspired by findings from cognitive linguistics, in particular the division of concepts expressed in language into two subsystems: the grammatical subsystem and the lexical subsystem [Talmy, 2000].[1] The lexical subsystem is expressed using so-called open-class elements (OCEs), i.e. nouns, verbs, adjectives and adverbs. Concepts associated with the grammatical subsystem are expressed using so-called closed-class elements (CCEs), including function words such as conjunctions, determiners, pronouns, and prepositions, but also suffixes such as plural markers and tense markers. Consider the following example, taken from [Evans and Pourcel, 2009], with CCEs printed in bold:

A *waiter serv****ed the*** *customer****s***

CCEs exhibit two important characteristics: First, *the inventory of CCEs is fixed*, i.e., whereas OCEs are constantly added to the language and vary enormously depending on the domain of the input text, the set of CCEs is limited, does not change over time and is the same for all domains. Due to the limited number of CCEs, finite CCE lists can be generated with fairly little effort for basically any language. Second, *CCEs occur very frequently*[2] *and provide a structuring function*, i.e. a 'scaffolding', across which concepts associated with the lexical subsystem can be draped [Evans and Pourcel, 2009].

We present a first version of a noun group chunker that makes use of this structuring function of CCEs. The general idea is to provide domain-independent information only (i.e. the CCE lists and a few linguistically-inspired seed rules) and make the system adapt itself automatically to the domain of the input text by bootstrapping domain-specific validation lists.

Based on the lists of CCEs and one seed extraction rule for each of the four OCE classes noun (N), verb (V), adjective (ADJ) and adverb (ADV), the system creates an initial validation set for each OCE class. A bootstrapping procedure is used to mutually expand the set of extraction rules and the validation sets in order to eventually assign one of the four OCE tags to all unknown (i.e. non-CCE) tokens in the input text. Based on the final tagging, sequences of ADJ and N tokens are extracted as noun groups.

The algorithm is described in detail in section 3. Evaluation results are presented in section 4.

2 Related Work

As our approach towards noun group extraction is based on the assignment of word class tags, our work is related to the task of part-of speech (POS) tagging. Unsupervised approaches to POS tagging usually disambiguate tags using a lexicon of possible tags for each token (e.g. [Merialdo, 1994], [Goldwater and Griffiths, 2007] and many others). However, these lexicons are large and, due to the openness of the OCE classes, can never be exhaustive. [Haghighi and Klein, 2006] replace the tagging lexicon by a prototype list, specifying three examples of each tag. We reduce the

[1] Note that Talmy considers this linguistic structuring as universal, i.e., it holds for any specific natural language. Hence, our approach reveals a high degree of language independency.

[2] Closed-class words constitute about 40% of an average English text [Höhle and Weissenborn, 1999].

used lexicon to the possible tags of CCEs to ensure domain-independency.

Our bootstrapping algorithm is similar to the procedures described by [Riloff and Jones, 1999] and [Collins and Singer, 1999] for labelling words with semantic categories. Starting with a small set of seed words representing each target semantic category and an unannotated corpus, [Riloff and Jones, 1999] create extraction patterns using syntactic templates, compute a score for each pattern based on the number of seed words among its extractions and use the best patterns to automatically label more words. Each newly labeled word is assigned a score based on how many different patterns extracted it and the best words remain in the semantic dictionary on which the next bootstrapping iteration is based. [Collins and Singer, 1999] use spelling rules in addition to contextual rules. [Yangarber *et al.*, 2002] present a bootstrapping approach to simultaneously learn diseases and locations and stress the usefulness of competing target categories. Similarly, we simultaneously learn extraction rules for all four OCE types.

3 Algorithm

3.1 CCE lists

Our CCE lexicon is based on the lists generated by [Spurk, 2006], to which we made some minor modifications. For example, we added a list of ordinal numbers (i.e. *first*, *second*, etc.), and introduced a special tag for the negation *not*. We also added a list of quantifiers used for grading, i.e *more*, *most*, *less*, *least*. These are used as part of the seed rule for extracting adjectives. We also removed Spurk's list of adverbs, which are strictly speaking OCEs.

3.2 General algorithm

The algorithm can be subdivided into four parts:

1. **Initialization**: Extract the initial validation sets based on the seed rules.
2. **Bootstrapping**: Iteratively expand the validation sets and the set of extraction rules and retag the input text.
3. **Postprocessing**: Tag all ambiguous and untagged tokens.
4. **Noun group extraction**: Extract noun groups based on the tagging.

Each of these parts is described in detail in the following sections.

3.3 Initialization

To initialize the bootstrapping process, we manually specified one seed rule for each of the four OCE types. The rules are listed in Table 1, where X represents any single non-CCE token, DET a determiner, PREP a preposition, PUNCT a punctuation symbol, BE some form of the auxiliary verb *be* and GRAD_ADV one of the four grading quantifier listed in section 3.1. The seed rule for adverbs makes use of the bound CCE *-ly*, with which adverbs are generated from adjectives. Based on the set of adjectives extracted using the adjective seed rule, we find adverbs by matching adjective seeds followed by *-ly*. Each rule extracts the part in bold as instance of the respective OCE type. It is assumed that these seed rules are trustworthy in the sense that the found matching elements for X are considered correct.

OCE type	*Seed rule*	*Example*
N	DET **X** PREP	the **computation** of
V	*to* **X** DET	to **give** the
ADJ	BE GRAD_ADV **X** PUNCT	is very **proud** .
ADV	**ADJ-ly**	**proudly**

Table 1: Seed rules

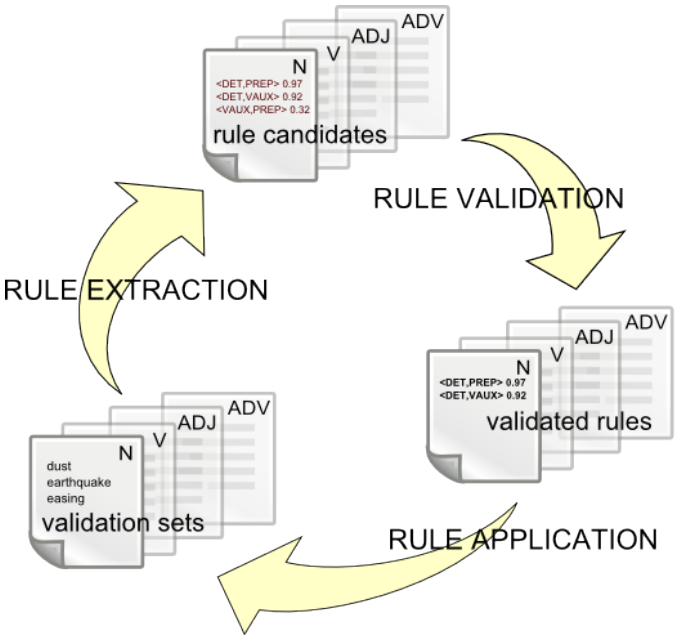


Figure 1: Bootstrapping loop

3.4 Bootstrapping

The bootstrapping loop is depicted in Figure 1. The three steps of each iteration are described in the following. The bootstrapping ends when no more rules are learnt.

Step 1: Rule Application

In the first iteration, we set the set of validated rules equal to the set of seed rules described in section 3.3. In the rule application step, the validated rules are applied to the input text by extracting all instances of X matching the respective pattern. For example, the seed rule <DET X PREP> for nouns extracts the seed *airport* from the sentence

> *Getting to and from <the:DET airport:X in:PREP> coming weeks may be the problem, however.*

The extracted instances are added to the validation set of the respective OCE class. For each extracted verb form, we also add other verb forms matched in the text, which are automatically generated based on the bound CCEs for verbs (i.e. *-(e)d*, *-s*, and *-ing*). For extracted adjectives, we also add the adverb built using *-ly* if found in the text.

After all rules have been applied, the input text is retagged based on the expanded validation sets. Tokens appearing in more than one validation list are assigned all possible tags, i.e. left ambiguous.

Step 2: Rule Extraction

New rule candidates are extracted using the validation sets generated in step 1. For each OCE class O, the method below is applied:

1. For each entry E in the validation set of O, match all <POS_L E POS_R> in the text, where POS_L (POS_R) correspond to the POS tag of the left (right) context of E, i.e. an already tagged token directly preceding (following) E.
2. Add <POS_L X POS_R> to the set of rule candidates for O.

Note that in the first iteration POS_L and POS_R represent some CCE tag. In later iterations, with some OCE tokens tagged based on the validation lists, POS_L and POS_R can also refer to an OCE tag.
For illustration of the rule extraxtion procedure, consider the following example. For the entry *airport* from the validation set of nouns, we can extract the rule candidate <DET X VAUX> from the sentence

> *While <the:DET airport:N was:VAUX> closed , flights were diverted.*

This rule is then put to the set of rule candidates for nouns.

Step 3: Rule Validation
The rule candidates extracted in step 2 are validated by calculating the accuracy of each rule candidate *r* for OCE *O* using formula 1,

$$acc(r) = \frac{pos_r + 1}{pos_r + neg_r + 1} \quad (1)$$

where pos_r refers to the number of occurrences matching the pattern <POS_L X_O POS_R>, neg_r refers to the number of occurrences matching the pattern <POS_L $X_{\neg O}$ POS_R>, and 1 is a smoothing constant. X_O refers to any token tagged with category O, $X_{\neg O}$ refers to any token tagged with any open class other than O.
If the calculated accuracy of a rule candidate exceeds a fixed threshold (currently set to 0.5), the rule candidate is added to the set of validated rules.
The input text is retagged based on the validated rules and again, ambiguous tokens are tagged with all possible tags.

3.5 Postprocessing

The postprocessing step serves two purposes: First, disambiguate all OCE tokens tagged with more than one tag. Second, tag all those tokens not appearing in any of the validation sets and not covered by any of the learned rules. To disambiguate OCE tokens to which more than one tag has been assigned, we compare the scores of all rules matching the context of the token and apply the highest-scoring one. Several matching rules are possible if the context of the token contains ambiguous tokens. For example, in order to tag the unknown token *August* in the sentence

> *Trade figures fail to show a substantial improvement from July <and:CONJ August:X 's:DET/VAUX> near-record deficits*,

we need to decide whether to apply the rule *<CONJ X DET>* (a verb rule) or *<CONJ X VAUX>* (a noun rule). As *score(<CONJ X VAUX>)* $= 0.94$ and *score(<CONJ X DET>)* $= 0.80$, we decide to apply the higher scoring noun rule and tag *August* as N.
To tag tokens that do not match any of the rules, we apply a backup procedure: We tag it based on its left context only. Here, we collect all rules with a matching left context, compute the average score of all rules for each of the tags in question, and assign the tag with the highest average score.
The postprocessing step is iterated until all tokens have been assigned a single tag.

3.6 Noun group extraction

After all tokens have been tagged, noun groups are collected by extracting all sequences of consecutive ADJ and N tokens. Note that all previous steps consider single tokens, not token sequences, i.e. a learned rule cannot be applied to extract a multi-word noun group directly. Instead, multi-word noun groups are extracted by learning and applying rules that involve OCE tags, e.g. the learned rule <DET X N> for tagging nouns, which tags *U.K.* as N in the sentence

> *But consumer expenditure data released Friday don't suggest that <the:DET U.K.:X economy:N> is slowing that quickly,*

given that *economy* has already been tagged as N.

4 Evaluation

The algorithm was evaluated on sections 15 to 18 of the Wall Street Journal corpus, a commonly used corpus for part-of speech tagging and chunking tasks, e.g. the CONLL 2000 shared task on chunking.[3] It contains 8,936 sentences with 46,874 noun groups (matching the regular expression $JJ^*(NNP|NN|NNS)^+$).
It is difficult to compare our system to others, which make use of more resources. The F-measure values of published results for the same dataset lie in the lower 90s, with a baseline F-measure of about 80 (cf. http://ifarm.nl/erikt/research/np-chunking.html). However, all these systems use a POS-annotated corpus as input, i.e., unlike our system, they require POS information to be available.
Due to the difficulty of comparison to other results, we decided to evaluate the system by taking a look at the learning process, and evaluate the effect of the initial seed rules as well as the bootstrapping and postprocessing procedures. As baseline, we tagged all non-CCE tokens with the most probable tag, N, thus extracting all chunks occurring between two CCEs as noun groups. We also evaluated the chunking result achieved using the initial validation sets extracted based on the four seed rules. Here, all tokens occurring in one of the initial validation sets were tagged accordingly, all other OCE tokens were tagged as N. In addition, we evaluated the final tagging, which was generated by applying all rules validated by the bootstrapping process as described in 3.5. The bootstrapping stopped after 7 iterations. All results are presented in Tables 2 and 3.
In the token-based evaluation, we look at the noun group tokens individually and evaluate how many of them are correctly considered part of a noun group by the algorithm. This evaluation procedure is similar to the one used for the CONLL 2000 shared task, which is also token-based. However, we do not evaluate based on the BIO tagging scheme, but count matching noun group tokens, irrespective of their position within the chunk.
The chunk-based evaluation is more strict in that it considers complete chunks only, i.e. if two of three tokens in a noun group have correctly been assigned an N tag, they are not counted as a match because one token is missing, i.e. the complete chunk was not recognized.

[3] http://www.cnts.ua.ac.be/conll2000/chunking/

	Precision	*Recall*	*F1*
Baseline	0.65	0.97	0.78
Initial tagging	0.68	0.96	0.79
Final Tagging	0.74	0.94	0.83

Table 2: Token-based evaluation of the bootstrapping

	Precision	*Recall*	*F1*
Baseline	0.50	0.66	0.57
Initial tagging	0.54	0.68	0.60
Final Tagging	0.60	0.72	0.66

Table 3: Chunk-based evaluation of the bootstrapping

5 Discussion and future work

We presented a first version of a self-adaptive noun chunker, which uses lists of closed-class elements, one seed extraction rule for each open class and a bootstrapping procedure to automatically generate and extend OCE validation sets and expand the set of extraction rules. An evaluation of the system's learning progress showed the usefulness of the bootstrapping procedure. The results are preliminary as we are presenting ongoing work, improvements are expected by optimizing the validation procedure. Currently, we only validate the rule candidates. Validating the extracted OCE tokens before adding them to the validation lists would make the algorithm more robust and prevent extraction errors from being propagated. In addition, the application of more sophisticated rule validation techniques, e.g. EM-based confidence estimation as described and used by [Jones, 2005] and [Tomita *et al.*, 2006], could improve the results.

In the current system, bound CCEs only play a minor role: When building additional verb forms for the extracted verbs and when building adverbs from adjectives. In the future, we also want to use bound CCEs to generate rules dealing with the morphology of the OCE tokens (i.e. add a second type of extraction rule, similar to the spelling features used by [Collins and Singer, 1999]).

We are currently evaluating the system on other, more specialized corpora in order to show its domain-independency. We also plan to evaluate it on texts in other languages. In addition, the influence of the size of the input text needs to be evaluated.

Acknowledgments

The work described in this paper has been carried out in the context of the research project DiLiA, co-funded by the European Regional Development Fund in the context of Investitionsbank Berlins ProFIT program (grant number: 10140159), and supported by a research grant from the German Federal Ministry of Economics and Technology to the project Theseus Ordo TechWatch (FKZ: 01MQ07016). We gratefully acknowledge this support.

References

[Collins and Singer, 1999] M. Collins and Y. Singer. Unsupervised models for named entity classification. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 100–111, Maryland, USA, 1999.

[Eichler and Neumann, 2010] K. Eichler and G. Neumann. DFKI KeyWE: Ranking keyphrases extracted from scientific articles. In *Proc. of the 5th International Workshop on Semantic Evaluations, ACL*, 2010.

[Evans and Pourcel, 2009] V. Evans and S. Pourcel, editors. *New Directions in Cognitive Linguistics*. Human Cognitive Processing. John Benjamins, 2009.

[Goldwater and Griffiths, 2007] S. Goldwater and T. Griffiths. A Fully Bayesian Approach to Unsupervised Part-of-Speech Tagging. In *Proc. of the ACL*, 2007.

[Haghighi and Klein, 2006] A. Haghighi and D. Klein. Prototype-Driven Learning for Sequence Models. In *Proc. of HLT/NAACL*, 2006.

[Höhle and Weissenborn, 1999] B. Höhle and J. Weissenborn. Discovering grammar. In A.D.Friederici and R. Menzel, editors, *Learning: Rule Extraction and Representation*. Walter de Gruyter, Berlin, 1999.

[Jones, 2005] Rosie Jones. *Learning to Extract Entities from Labeled and Unlabeled Texts*. PhD thesis, School of Computer Science, Carnegie Mellon University, 2005.

[Merialdo, 1994] B. Merialdo. Tagging english text with a probabilistic model. *Computational Linguistics*, 20(2):155–171, 1994.

[Riloff and Jones, 1999] E. Riloff and R. Jones. Learning dictionaries for information extraction by multi-level bootstrapping. In *Proceedings of the 16th National Conference on AI (AAAI-99)*, Orlando, FL, 1999.

[Spurk, 2006] C. Spurk. Ein minimal berwachtes Verfahren zur Erkennung generischer Eigennamen in freien Texten. Diplomarbeit, Saarland University, Germany, 2006.

[Talmy, 2000] L. Talmy. *Towards a cognitive semantics*. MIT Press, Cambridge, MA, 2000.

[Tomita *et al.*, 2006] J. Tomita, S. Soderland, and O. Etzioni. Expanding the recall of relation extraction by bootstrapping. In *EACL Workshop on Adaptive Text Extraction and Mining*, 2006.

[Yangarber *et al.*, 2002] R. Yangarber, L. Winston, and R. Grishman. Unsupervised learning of generalized names. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING 2002)*, 2002.

Towards Adjusting Mobile Devices to User's Behaviour

Fricke, P. [*], **Jungermann, F.**[*], **Morik, K.**[*], **Piatkowski, N.**[*], **Spinczyk, O.**[†], and **Stolpe, M.**[*]
Technical University of Dortmund

Abstract

Mobile devices are a special class of resource-constrained embedded devices. Computing power, memory, the available energy, and network bandwidth are often severely limited. These constrained resources require extensive optimization of a mobile system compared to larger systems. Any needless operation has to be avoided. Time-consuming operations have to be started early on. For instance, loading files ideally starts before the user wants to access the file. So-called prefetching strategies optimize system's operation. Our goal is to adjust such strategies on the basis of logged system data. Optimization is then achieved by predicting an application's behavior based on facts learned from earlier runs on the same system. In this paper, we analyze system-calls on operating system level. The learned model predicts if a system-call is going to open a file fully, partially, or just for changing its rights.

1 Introduction

Users demand mobile devices to have long battery life, short application startup time, and low latencies. Mobile devices are constrained in computing power, memory, energy, and network connectivity. This conflict between user expectations and resource constraints can be reduced, if we tailor a mobile device such that it uses its capacities carefully for exactly the user's needs, i.e., the services, that the user wants to use. Predicting the user's behavior given previous behavior is a machine learning task. For example, based on the learning of most often used file path components, a system may avoid unnecessary probing of files and could intelligently prefetch files. Prefetching those files, which soon will be read by the user, leads to decreased startup latencies for applications and, accordingly, conservation of energy.

The resource restrictions of mobile devices motivate the application of machine learning for predicting user behavior. At the same time, machine learning dissipates resources. There are three critical resource constraints:

- Data gathering: logging user actions uses processing capacity.
- Data storage: the training and test data as well as the learned model use memory.
- Communication: if training and testing is performed on a central server, sending data and the resulting model uses the communication network.
- Response time: the prediction of usage, i.e., the model application, has to happen in short real-time.

The dilemma of saving resources at the device through learning which, in turn, uses up resources, can be solved in several ways. Here, we set aside the problem of data gathering and its prerequisites on behalf of operation systems for embedded systems [Lohmann *et al.*, 2009] [Tartler *et al.*,] [Cantrill *et al.*, 2004].This is an important issue in its own right. Regarding the other restrictions, especially the restriction of memory, leads us to two alternatives.

Server-based learning: The learning of usage profiles from data is performed on a server and only the resulting model is communicated back to the device. Learning is less restricted in runtime and memory consumption. Just the learned model must obey the runtime and communication restrictions. Hence, a complex learning method is applicable. Figure 1 shows this alternative.

Device-based learning: The learning of usage profiles on the device is severely restricted in complexity. It does not need any communication but requires training data to be stored. Data streaming algorithms come into play in two alternative ways. First, descriptive algorithms incrementally build-up a compact way to store data. They do not classify or predict anything. Hence, in addition, simple methods are needed that learn from the aggregated compact data. Second, simple online algorithms predict usage behavior in realtime. The latter option might only be possible if specialized hardware is used, e.g., General Purpose GPUs. Figure 2 shows this alternative.

In this paper, we want to investigate the two alternatives using logged system calls. Server-based learning is exemplified by predicting file-access types in order to enhance prefetching. It is an open question whether structural models are demanded for the prediction of user behavior on the basis of system calls, or simpler models such as Naive Bayes suffice. Should the sequential nature of system calls be taken into account by the algorithm? Or is it sufficient to encode the sequences into the features? Or should features as well as algorithm be capable of explicitly addressing se-

[*] Artificial Intelligence Group
Baroper Strasse 301, Dortmund, Germany
{fricke,jungermann,morik,piatkowski,stolpe}@ls8.cs.tu-dortmund.de

[†] Embedded System Software Group
Otto-Hahn-Strasse 16, Dortmund, Germany
olaf.spinczyk@tu-dortmund.de

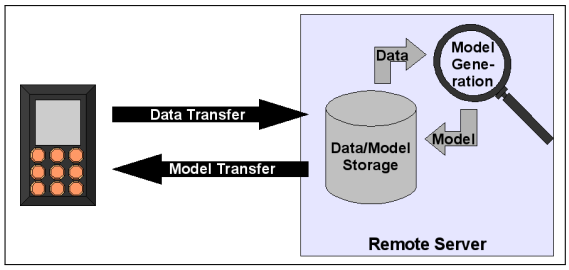


Figure 1: Server-based Architecture

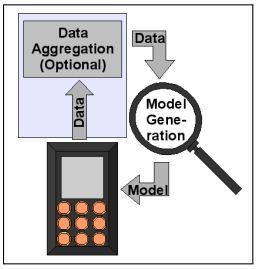


Figure 2: Device-based Architecture

quences? We investigate the use of two extremes, Conditional Random Fields (CRF) and Naive Bayes (NB). In particular, we inspect their memory consumption and runtime, both, for training and applying the learned function. Section 2 presents the study of server-based learning for ubiquitous devices. We derive the learning task from the need of enhancing prefetching strategies, describe the log data used, and present the learning results together with resource consumptions of NB and CRF.

Device-based learning is exemplified by recognizing applications from system calls in order to prevent fraud. We apply the data streaming algorithm Hierarchical Heavy Hitters (HHH) yielding a compact data structure for storage. Using these, the simple kNN method classifies systems calls. In particular, we investigate how much HHH compress data. Section 3 presents the study of device-based learning using a streaming algorithm for storing compact data. We conclude in Section 4 by indicating related and future work.

2 Server-based Learning

In this section we present the first case-study, where log data are stored and analyzed on a server (data are described in Section 2.2). Learning aims at predicting file access in order to prefetch files (see Section 2.1). The learning methods NB and CRF are introduced shortly in Section 2.3 and Section 2.4, respectively. The results are shown in Section 2.5.

2.1 File-access pattern prediction

A prediction of file-access patterns is of major importance for the performance and resource consumption of system software. For example, the Linux operating system uses a large "buffer cache" memory for disk blocks. If a requested disk block is already stored in the cache (*cache hit*), the operating system can deliver it to the application much faster and with less energy consumption than otherwise (*cache miss*). In order to manage the cache the operating system has to implement two strategies, block replacement and prefetching. The *block replacement* strategy is consulted upon a cache miss: a new block has to be inserted into the cache. If the cache is already full, the strategy has to decide which block has to be replaced. The most effective victim is the one with the longest forward distance, i.e. the block with the maximum difference between now and the time of the next access. This requires to know or guess the future sequence of cache access. The *prefetching* strategy proactively loads blocks from disk into the cache, even if they have not been requested by an application, yet. This often pays off, because reading a bigger amount of blocks at once is more efficient than multiple read operations. However, prefetching should only be performed if a block will be needed in the near future. For both strategies, block replacement and prefetching, a good prediction of future application behavior is crucial.

Linux and other operating systems still use simple heuristic implementations of the buffer cache management strategies. For instance, the prefetching code in Linux [Bovet and Cesati, 2005] continuously monitors read operations. As long as a file is accessed sequentially the read ahead is increased. Certain upper and lower bounds restrict the risk of mispredictions. This heuristics has two flaws:

- No prefetching is performed *before* the first read operation on a specific file, e.g., after "open", or even earlier.
- The strategy is based on assumptions on typical disk performance and buffer cache sizes, in general. However, these assumptions might turn out to be wrong in certain application areas or for certain users.

Prefetching based on machine learning avoids both problems. Prefetching can already be performed when a file is opened. It only depends on the prediction that the file will be read. The prediction is based on empirical data and not on mere assumptions. If the usage data change, the model changes, as well.

2.2 System Call Data for Access Prediction

We logged streams of system calls of type FILE, which consist of various typical sub-sequences, each starting with an `open`- and terminating with a `close`-call, like those shown in Figure 3.We collapsed such sub-sequences to one observation and assign the class label

- **full**, if the opened file was read from the first seek (if any) to the end,
- **read**, if the opened file was randomly accessed and
- **zero**, if the opened file was not read after all.

We propose the following generalization of obtained filenames. If a file is regular, we remove anything except the filename extension. Directory names are replaced by "DIR", except for paths starting with "/tmp" – those are replaced by "TEMP". Any other filenames are replaced by "OTHER". This generalization of filenames yields good results in our experiments. Volatile information like thread-id, process-id, parent-id and system-call parameters is dropped, and consecutive observations are compound to one sequence if they belong to the same process. The resulting dataset consists of 673887 observations in 80661 sequences, a snippet[1] is shown in Table 1.

We used two feature sets for the given task. The first encodes information about sequencing as features, resulting in 24 features, namely f_t, f_{t-1}, f_{t-2}, f_{t-2}/f_{t-1}, f_{t-1}/f_t,

[1]The final dataset is available at:
`http://www-ai.cs.tu-dortmund.de/PUBDOWNLOAD/MUSE2010`

```
1,open,1812,179,178,201,200,eclipse,/etc/hosts,524288,438,7 : 361, full
2,read,1812,179,178,201,200,eclipse,/etc/hosts,4096,361
3,read,1812,179,178,201,200,eclipse,/etc/hosts,4096,0
4,close,1812,179,178,201,200,eclipse,/etc/hosts
```

Figure 3: A sequence of system calls to *read* a file. The data layout is: timestamp, syscall, thread-id, process-id, parent, user, group, exec, file, parameters (optional) : read bytes, label (optional)

user	group	exec	file	label
20005	10000	firefox-bin	cookies.sqlite-journal	zero
20005	10000	firefox-bin	default	zero
20005	10000	firefox-bin	hosts	full
20005	10000	firefox-bin	hosts	full
20005	10000	multiload-apple	mtab	full
10028	10000	kmail	png	zero

Table 1: Snippet of the final dataset.

predicted\true	full	zero	read
full	0	2	1
zero	5	0	4
read	4	2	0

Table 2: Cost matrix

$f_{t-2}/f_{t-1}/f_t$, with $f \in \{user, group, exec, file\}$. The second feature set simply uses two features $exec_{t-1}/exec_t$ and $file_{t-2}/file_{t-1}/file_t$ as its only features.

Errors in predicting the types of access result in different degrees of failure. Predicting a partial caching of a file, if just the rights of a file have to be changed, is not as problematic as predicting a partial read if the file is to be read completely. Hence, we define a cost-matrix (see Table 2) for the evaluation of our approach.

2.3 Naive Bayes Classifier

The Naive Bayes classifier [Hastie *et al.*, 2003] assigns labels $y \in Y$ to examples $x \in X$. Each example is a vector of m attributes written here as x_i, where $i = 1...m$. The probability of a label given an example is according to the Bayes Theorem:

$$p(Y|x_1, x_2, ..., x_m) = \frac{p(Y)p\left(x_1, x_2, ..., x_m|Y\right)}{p\left(x_1, x_2, ..., x_m\right)} \quad (1)$$

Domingos and Pazzani [Domingos and Pazzani, 1996] rewrite eq. (1) and define the *Simple Bayes Classifier* (SBC):

$$p(Y|x_1, x_2, ..., x_m) = \frac{p(Y)}{p\left(x_1, x_2, ..., x_m\right)} \prod_{j=1}^{n} p\left(x_j|Y\right) \quad (2)$$

The classifier delivers the most probable class Y for a given example $x = x_1 \dots x_m$:

$$\arg\max_Y p(Y|x_1, x_2, ..., x_m) = \frac{p(Y)}{p\left(x_1, x_2, ..., x_m\right)} \prod_{j=1}^{m} p\left(x_j|Y\right) \quad (3)$$

The term $p\left(x_1, x_2, ..., x_m\right)$ can be neglected in eq. (3) because it is a constant for every class $y \in Y$. The decision for the most probable class y for a given example x just depends on $p(Y)$ and $p\left(x_i|Y\right)$ for $i = 1 \dots m$. These probabilities can be calculated after one run on the training data. So, the training runtime is $\mathcal{O}(n)$, where n is the number of examples in the training set. The number of probabilities to be stored during training are $|\mathcal{Y}| + (\sum_{i=1}^{m} |\mathcal{X}_j| * |\mathcal{Y}|)$, where $|\mathcal{Y}|$ is the number of classes and $|\mathcal{X}_i|$ is the number of different values of the ith attribute. The storage requirements for the trained model are $\mathcal{O}(mn)$.

It has often been shown that SBC or NBC perform quite well for many data mining tasks [Domingos and Pazzani, 1996; Huang *et al.*, 2003; Frank and Asuncion, 2010].

2.4 Linear-chain Conditional Random Fields

Linear-chain Conditional Random Fields, introduced by Lafferty et al. [Lafferty *et al.*, 2001], can be understood as discriminative, sequential version of Naive Bayes Classifiers. The conditional probability for an actual sequence of labels $\mathbf{y_1}, \mathbf{y_2}, ..., \mathbf{y_m}$, given a sequence of observations $\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_m}$ is modeled as an exponential family. The underlying assumption is that a class label at the current timestep t just depends on the label of its direct ancestor, given the observation sequence. Dependency among the observations is not explicitly represented, which allows the use of rich, overlapping features. Equation 4 shows the model formulation of linear-chain CRF

$$p_\lambda\left(Y = \mathbf{y}|X = \mathbf{x}\right) = \frac{1}{Z\left(\mathbf{x}\right)} \prod_{t=1}^{T} \exp\left(\sum_k \lambda_k f_k\left(y_t, y_{t-1}, \mathbf{x}\right)\right) \quad (4)$$

with the observation-sequence dependent normalization factor

$$Z\left(\mathbf{x}\right) = \sum_{\mathbf{y}} \prod_{t=1}^{T} \exp\left(\sum_k \lambda_k f_k\left(y_t, y_{t-1}, \mathbf{x}\right)\right) \quad (5)$$

The sufficient statistics or *feature functions* f_k are most often binary indicator functions which evaluate to 1 only for a single combination of class label(s) and attribute value. The parameters λ_k can be regarded as weights or scores for this feature functions. In linear-chain CRF, each attribute value usually gets $|\mathcal{Y}| + |\mathcal{Y}|^2$ parameters, that is one score per state-attribute pair as well as one score for every transition-attribute triple, which results in a total of $\sum_{i=1}^{m} |\mathcal{X}_i| \left(|\mathcal{Y}| + |\mathcal{Y}|^2\right)$ model parameters, where $|\mathcal{Y}|$ is the number of classes, m is the number of attributes and $|\mathcal{X}_i|$ is the number of different values of the ith attribute. Notice that the feature functions explicitly depend on the whole observation-sequence rather than on the attributes at time t. Hence, it is possible and common to involve attributes of preceding as well as following observations from the current sequence into the computation of the total score

$\exp\left(\sum_k \lambda_k f_k\left(y_t, y_{t-1}, \mathbf{x}\right)\right)$ for the transition from y_{t-1} to y_t given $\mathbf{x}$.

The parameters are usually estimated by the maximum-likelihood method, i.e., maximizing the conditional likelihood (Eq. 6) by quasi-Newton [Malouf, 2002], [Sha and Pereira, 2003], [Nocedal, 1980] or stochastic gradient methods [Vishwanathan *et al.*, 2006], [Schraudolph and Graepel, 2002], [Schraudolph *et al.*, 2007].

$$\mathcal{L}\left(\lambda\right) = \prod_{i=1}^{N} p_\lambda(Y = \mathbf{y}^{(i)}|X = \mathbf{x}^{(i)}) \tag{6}$$

The actual class prediction for an unlabeled observation-sequence is done by the Viterbi algorithm known from Hidden Marcov Models [Sutton and McCallum, 2007], [Rabiner, 1989].

Although CRF in general allow to model arbitrary dependencies between the class labels, efficient exact inference can solely be done for linear-chain CRF. This is no problem here, because they match the sequential structure of our system-call data, presented in section 2.2.

2.5 Results of Server-based Prediction

Comparing the prediction quality of the simple NB models and the more complex CRF models, surprisingly, the CRF are only slightly better when using the two best features (see Tables 3 and 5). CRF outperforms NB when using all features (see Tables 4 and 6). These two findings indicate that the sequence information is not as important as we expected. Neither encoding the sequence into features nor applying an algorithm which is made for sequential information outperforms a simple model. The Tables show that precision, recall, accuracy, and misclassification cost are quite homogeneous for CRF, but vary for NB. In particular, the precision of predicting "read" and the recall of class "zero" differs from the numbers for the other classes, respectively. This makes CRF more reliable.

Inspecting resource consumption, we stored models of the two methods for both feature sets and for various numbers of examples to show the practical storage needs of the methods.Table 9 presents the model sizes of the naive Bayes classifier on both feature sets and for various example set sizes. We used the popular open source data mining tool *RapidMiner*[2] for these experiments. Table 9 also shows the model sizes of CRF on both feature sets and various example set sizes.

We used the open source CRF implementation *CRF++*[3] with L_2-regularization, $\sigma = 1$ and L-BFGS optimizer in all CRF experiments. Obviously, the storage needs for a model produced by a NB classifier are lower than those for a CRF model. This is the price to be paid for more reliable prediction quality. CRF don't scale-up well. Considering training time, the picture becomes worse. Table 10 shows the training time of linear-chain or HMM-like CRF consuming orders of magnitude more time than NB.

predicted\true	full	zero	read	prec.
full	1427467	19409	3427	98.43
zero	12541	2469821	40258	97.91
read	80872	217380	2467695	89.22
recall	93.86	91.25	98.26	

Table 3: Result of Naive Bayes Classifier on best two features, 10x10-fold cross-validated, accuracy: 94.45 $\pm$ 0.00, missclassification costs: 0.152 $\pm$ 0.001

full	zero	read	prec.
1426858	21562	22717	96.99
15392	2371009	97566	95.45
78630	314039	2391097	85.89
93.82	87.60	95.21	

Table 4: Result of Naive Bayes Classifier on all 24 features, 10x10-fold cross-validated, accuracy: 91.84 $\pm$ 0.00, missclassification costs: 0.218 $\pm$ 0.002

predicted\true	full	zero	read	prec.
full	1446242	7123	29051	97.56
zero	19452	2639097	133007	94.54
read	55186	60390	2349322	95.31
recall	95.09	97.51	93.55	

Table 5: Result of HMM-like CRF on the best two features, 10x10-fold cross-validated, accuracy: 95.49 $\pm$ 0.00, missclassification costs: 0.150 $\pm$ 0.000

full	zero	read	prec.
1450147	8335	25629	97.71
14563	2639724	126403	94.93
56170	58551	2359348	95.36
95.35	97.53	93.95	

Table 6: Result of HMM-like CRF on all 24 features, 10x10-fold cross-validated, accuracy: 95.70 $\pm$ 0.00, missclassification costs: 0.143 $\pm$ 0.000

predicted\true	full	zero	read	prec.
full	1467440	4733	7503	99.17
zero	10883	2659294	108340	95.71
read	42557	42583	2395537	96.57
recall	96.49	98.25	95.39	

Table 7: Result of linear-chain CRF on the best two features, 10x10-fold cross-validated, accuracy: 96.79 $\pm$ 0.00, missclassification costs: 0.112 $\pm$ 0.000

full	zero	read	prec.
1468095	4117	5022	99.38
10306	2662966	107859	95.75
42479	39527	2398499	96.69
96.53	98.39	95.51	

Table 8: Result of linear-chain CRF on all 24 features, 10x10-fold cross-validated, accuracy: 96.89 $\pm$ 0.00, missclassification costs: 0.110 $\pm$ 0.000

3 Device-based Learning

In this section, we present the second case-study, where streams of log data are processed in order to store patterns of system use. The goal is to aggregate the streaming system data. A simple learning method might then use the aggregated data. The method of Hierarchical Heavy Hitters (HHH) is defined in Section 3.1. The log data are shown in Section 3.2. For the comparison of different sets of HHH, we present a distance measure that allows for clustering or classifying sets of HHH. In addition to the quality of our HHH application, its resource consumption is presented in Section 3.3.

3.1 Hierarchical Heavy Hitters

The *heavy hitter problem* consists of finding all frequent elements and their frequency values in a data set. According

[2]RapidMiner is available at:
`http://www.rapidminer.com`

[3]CRF++ is available at:
`http://crfpp.sourceforge.net/`

#Att.\#Seq.	0	67k	135k	202k	270k	337k	404k	472k	539k	606k	674k
2 nB	2	78	100	118	132	143	154	161	169	176	181
24 nB	5	247	310	355	392	417	448	469	488	505	517
2 CRF++ (HMM)	5	247	366	458	490	512	569	592	614	634	649
24 CRF++ (HMM)	12	615	878	1102	1170	1216	1367	1420	1463	1521	1551
2 CRF++	6	523	776	978	1043	1089	1213	1260	1299	1345	1378
24 CRF++	19	1339	1914	2415	2559	2652	2988	3095	3184	3303	3365

Table 9: Storage needs (in kB) of the naive Bayes (nB), the HMM-like CRF (CRF++ (HMM)) and the linear-chain CRF (CRF++) classifier model on different numbers of sequences and attributes.

#Att.\#Seq.	0	67k	135k	202k	270k	337k	404k	472k	539k	606k	674k
2 nB	< 1	< 1	< 1	< 1	1	< 1	< 1	< 1	< 1	< 1	< 1
24 nB	< 1	< 1	< 1	1	< 1	1	1	1	1	2	1
2 CRF++ (HMM)	< 1	9.09	28.56	44.08	60.1	75.76	107.28	127.04	149.95	165.94	199.2
24 CRF++ (HMM)	< 1	27.92	55.9	103.24	153.53	160.33	230.7	273.29	232.84	309.19	317.62
2 CRF++	< 1	16.69	50.23	85.18	113.21	145.96	173.56	200.98	234.65	260.56	325.54
24 CRF++	< 1	41.06	105.29	156.67	296.31	300.83	343.28	433.03	440.88	463.84	632.96

Table 10: Training time (in seconds) of the naive Bayes (nB), the HMM-like CRF (CRF++ (HMM)) and the linear-chain CRF (CRF++) classifier model on different numbers of sequences and attributes.

to Cormode [Cormode *et al.*, 2003], given a (multi)set S of size N and a threshold $0 < \phi < 1$, an element e is a *heavy hitter* if its frequency $f(e)$ in S is not smaller than $\lfloor \phi N \rfloor$. The set of heavy hitters is then $HH = \{e | f(e) \geq \lfloor \phi N \rfloor\}$.

If the elements in S originate from a hierarchical domain D, one can state the following problem [Cormode *et al.*, 2003]:

Definition 1 (HHH Problem) *Given a (multi)set S of size N with elements e from a hierarchical domain D of height h, a threshold $\phi \in (0, 1)$ and an error parameter $\epsilon \in (0, \phi)$, the* Hierarchical Heavy Hitter Problem *is that of identifying prefixes $P \in D$, and estimates f_p of their associated frequencies, on the first N consecutive elements S_N of S to satisfy the following conditions:*

- *accuracy: $f_p^* - \varepsilon N \leq f_p \leq f_p^*$, where f_p^* is the true frequency of p in S_N.*
- *coverage: all prefixes $q \notin P$ satisfy $\phi N > \sum f(e) : (e \preceq q) \wedge (\nexists p \in P : e \preceq p)$.*

Here, $e \preceq p$ means that element e is *generalizable* to p (or $e = p$). For the extended multi-dimensional heavy hitter problem introduced in [Cormode *et al.*, 2004], elements can be multi-dimensional d-tuples of hierarchical values that originate from d different hierarchical domains with depth $h_i, i = 1, \ldots, d$. There exist two variants of algorithms for the calculation of multi-dimensional HHHs: Full Ancestry and Partial Ancestry, which we have both implemented. For a detailed description of these algorithms, see [Cormode *et al.*, 2008].

3.2 System Call Data for HHH

The kernel of current Linux operating systems offers about 320 different types of system calls to developers. Having gathered all system calls made by several applications, we observed that about 99% of all calls belonged to one of the 54 different call types shown in Table 11. The functional categorization of system calls into five groups is due to [Silberschatz *et al.*, 2010]. We focus on those calls only, since the remaining 266 call types are contained in only 1% of the data and therefore can't be frequent.

HHHs can handle values that have a hierarchical structure. We have utilized this expressive power by representing system calls as tuples of up to three hierarchical feature values. Each value originates from a taxonomy (*type*, *path* or *sequence*) that either can be derived dynamically from the data itself or has to be defined explicitly by the user. The groups introduced in Table 11 form the top level of the taxonomy for the hierarchical variable *type* (see Fig. 4). The `socket` call is a child of group COMM and FILE is the parent of calls like `open` and `fcntl64`. Subtypes of system calls can be defined by considering the possible values of their parameters. For example, the `fcntl64` call which operates on file descriptors has *fd*, *cmd* and *arg* as its parameters. We have divided the 16 different nominal values of the *cmd* parameter into seven groups — `notify`, `dflags`, `duplicate`, `sig`, `lock`, `fflags` and `lease` — that have become the children of the `fcntl64` system call in our taxonomy (see Fig. 4). One may further divide `fcntl64` calls of subtype `fflags` by the values `F_SETFL` and `F_GETFL` of the *arg* parameter. In the same way, we defined parents and children for each of the 54 call types and their parameters.

FILE	COMM	PROC	INFO	DEV
`open`	`recvmsg`	`mmap2`	`access`	`ioctl`
`read`	`recv`	`munmap`	`getdents`	
`write`	`send`	`brk`	`getdents64`	
`lseek`	`sendmsg`	`clone`	`clock_gettime`	
`_llseek`	`sendfile`	`fork`	`gettimeofday`	
`writev`	`sendto`	`vfork`	`time`	
`fcntl`	`rt_sigaction`	`mprotect`	`uname`	
`fcntl64`	`pipe`	`unshare`	`poll`	
`dup`	`pipe2`	`execve`	`fstat`	
`dup2`	`socket`	`futex`	`fstat64`	
`dup3`	`accept`	`nanosleep`	`lstat`	
`close`	`accept4`		`lstat64`	
			`stat`	
			`stat64`	
			`inotify_init`	
			`inotify_init1`	
			`readlink`	
			`select`	

Table 11: We focus on 54 system call types which are functionally categorized into five groups. FILE: file system operations, COMM: communication, PROC: process and memory management, INFO: informative calls, DEV: operations on devices.

The *path* variable is filled whenever a system call accesses a file system path. Its hierarchy comes naturally along with the given path hierarchy of the file system. The *sequence* variable expresses the temporal order of calls within a process. The directly preceding call is the highest, less recent calls are at deeper levels of the hierarchy.

We collected system call data from eleven applications

		Memory			Run-time			Similarity	
		Min	Max	Avg	Min	Max	Avg	Avg	Dev
FA	T	19	151	111	16	219	79	**0.997**	0.006
	TP	25	9,971	5,988	31	922	472	0.994	0.003
	TPS	736	**73,403**	48,820	78	14,422	6,569	0.987	0.008
PA	T	7	105	70	15	219	74	**0.985**	0.010
	TP	7	4,671	2,837	31	5,109	2,328	0.957	0.017
	TPS	141	18,058	10,547	78	**150,781**	74,342	0.921	0.026

Table 12: Memory consumption (number of stored tupels), run-time (milliseconds) and similarity to exact solution of the Full Ancestry (FA) and Partial Ancestry (PA) algorithms (ε = 0.0005, ϕ = 0.002). Minimum (Min), maximum (Max) and average (Avg) values were calculated over measurements for the first log file of all eleven applications with varying dimensionality of the element tupels (T = *type* hierarchy, P = *path* hierarchy, S = *sequence* hierarchy).

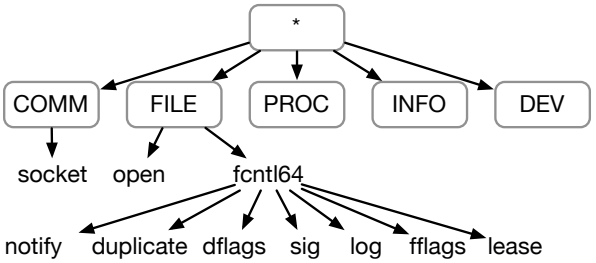


Figure 4: Parts of the taxonomy we defined for the hierarchical variable *type*.

(like Firefox, Epiphany, NEdit, XEmacs) with the *strace* tool (version 4.5.17) under Ubuntu Linux (kernel 2.6.26, 32 bit). All child processes were monitored by using option `-f` of *strace*. For each application, we logged five times five minutes and five times ten minutes of system calls if they belonged to one of the 54 types shown in Table 11, resulting in a whole of 110 log files comprising about 23 million of lines (1.8 GB).

3.3 Resulting Aggregation through Hierarchical Heavy Hitters

We have implemented the Full Ancestry and Partial Ancestry variants of the HHH algorithm mentioned in Section 3.1. The code was integrated into the RapidMiner data mining tool. Regarding run-time, all experiments were done on a machine with Intel Core 2 Duo E6300 processor with 2 GHz and 2 GB main memory.

Since we want to aggregate system call data on devices that are severely limited in processing power and available memory, measuring the resource usage of our algorithms was of paramount importance. Table 12 shows the run-time and memory consumption of the Full Ancestry and Partial Ancestry algorithms using only the *type* hierarchy, the *type* and *path* hierarchy, or the *type*, *path*, and *sequence* hierarchy. Minimum, maximum and averages were calculated over a sample of the ten gathered log files for each of the eleven application by taking only the first log file for each application into account.

Memory consumption and run-time increase with the dimensionality of the elements, while at the same time approximation quality decreases. Quality is measured as similarity to the exact solution. Full Ancestry has a higher approximation quality in general. The results correspond to observations made by Cormode and are probably due to the fact that Partial Ancestry outputs bigger HHH sets, which was the case in our experiments, too. Note that approximation quality can always be increased by changing parameter ε to a smaller value at the expense of a longer run-time.

Even for three-dimensional elements, memory consumption is quite low regarding the number of stored tuples. The largest number of tuples, 73,403, only equates to a few hundred kilobytes in main memory! The longest run-time of 150,781 ms for Partial Ancestry in three dimensions relates to the size of the biggest log file (application Rhythmbox).

Figure 5 shows the behaviour of our algorithms on the biggest log file (application Rhythmbox) for three dimensions with varying ε and constant ϕ. Memory consumption and quality decrease with increasing ε, while the run-time increases. So the most important trade-off involved here is weighting memory consumption against approximation quality — the run-time is only linearly affected by parameter ε. Again, Full Ancestry shows a better approximation quality in general.

Classification results

For the 110 log files of all applications, we determined the HHHs, resulting in sets of frequent tupels of hierarchical values. Interpreting each HHH set as an example of application behaviour, we wanted to answer the question if the profiles could be separated by a classifier. So we estimated the expected classification performance by a leave-one-out validation for kNN.

Therefore, we needed to define a distance measure for the profiles determined by HHH algorithms. The data structures of HHH algorithms contain a small subset of prefixes of stream elements. The estimated frequencies f_p are calculated from such data structure by the output method and compared to ϕ, thereby generating a HHH set. The similarity measure DSM operates not on the HHH sets, but directly on the internal data structures D_1, D_2 of two HHH algorithms:

$$\mathrm{sim}(D_1, D_2) = \frac{\sum_{p \in P_1 \cap P_2} \mathrm{contrib}_{\mathrm{DSM}}(p)}{|P_1 \cup P_2|} .$$

Be f_p^i the estimated frequency of prefix p for data structure D_i as normally calculated by the HHH output method. The contribution of individual prefixes to overall similarity can then be defined as

$$\mathrm{contrib}_{\mathrm{DSM}}(p) = \frac{2 \cdot \min(f_p^1, f_p^2)}{\min(f_p^1, f_p^2) + \max(f_p^1, f_p^2)} .$$

The so defined similarity measure is independent from the choice of ϕ, as no HHH sets need to be calculated.

The classification errors for different values of k, hierarchies and distance measures are shown in Table 13. The new DSM distance measure which is independent of parameter ϕ shows the lowest classification error in all validation experiments. As a baseline, we also determined the relative frequencies (TF, term frequencies) of call types per

	T		TS	
k	DSM	TF	DSM	TF
3	**10.3**	17.0	**7.7**	17.0
5	**12.7**	18.7	**8.7**	18.7
7	**14.0**	21.7	**8.7**	21.7
9	**14.0**	21.0	**9.0**	21.0

Table 13: Results for kNN (k = 3, 5, 7, 9), $\varepsilon = 0.0005$, $\phi = 0.002$ and distance measures DSM and TF, when only the *type* hierarchy or *type* and *sequence* hierarchy together are used.

log file and classified them using kNN (with Euclidean distance). The error for profiling by HHH sets is significantly lower than for the baseline.

4 Conclusion

Server-based and device-based learning has been investigated regarding resource constraints. memory consumption. Aggregation using HHH worked successfully for the classification of applications. Further work will exploit HHH aggregation for other learning tasks and inspect other data streaming algorithms. Concerning server-based learning, we may now answer the questions from the introduction, whether structural models are demanded for the prediction of user behavior on the basis of system calls, or simpler models such as Naive Bayes suffice. Should the sequential nature of system calls be taken into account by the algorithm? Or is it sufficient to encode the sequences into the features? Or should features as well as algorithm be capable of explicitly addressing sequences? We have compared CRF and NB with respect to their model quality, memory consumption, and runtime. Neither encoding the sequence into features nor applying an algorithm which is made for sequential information (i.e., CRF) outperforms a simple model (i.e., NB).

This is in contrast with studies on intrusion detection, where it was shown advantageous to take into account the structure of system calls, utilizing Conditional Random Fields (CRF) [Gupta *et al.*, 2007] and special kernel functions to measure the similarity of sequences [Tian *et al.*, 2007]. Structured models in terms of special tree kernel functions outperformed n-gram representations when detecting malicious SQL queries [Bockermann *et al.*, 2009]. Possibly, for prefetching strategies, the temporal order of system calls is not as important as we expected it to be. In the near future the resulting improvements in terms of cache hit rate and file operation latencies will be evaluated systematically based on a cache simulator and by modifying the Linux kernel.

Given regular processors, CRF are only applicable in server-based learning. Possibly, the integration of special processors into devices and a massively parallel training algorithm could speed up CRF for device-based learning. Further work will implement CRF on a GPGPU (general purpose graphic processing unit). GPGPUs will soon be used by mobile devices. It has been shown that their energy efficiency is advantagous [Timm *et al.*, 2010].

References

[Bockermann *et al.*, 2009] Christian Bockermann, Martin Apel, and Michael Meier. Learning sql for database intrusion detection using context-sensitive modelling. In *Proc. 6th Detection of Intrusions and Malware, and Vulnerability Assessment*, pages 196 – 205. Springer, 2009.

[Bovet and Cesati, 2005] Daniel Bovet and Marco Cesati. *Understanding the Linux Kernel, Third Edition*. O'Reilly & Associates, Inc., Sebastopol, CA, USA, 2005.

[Cantrill *et al.*, 2004] Bryan M. Cantrill, Michael W. Shapiro, and Adam H. Leventhal. Dynamic instrumentation of production systems. In *Proc. of USENIX ATEC '04*, Berkeley, USA, 2004. USENIX.

[Cormode *et al.*, 2003] Graham Cormode, Flip Korn, S. Muthukrishnan, and Divesh Srivastava. Finding hierarchical heavy hitters in data streams. In *VLDB '2003: Proceedings of the 29th international conference on Very large data bases*, pages 464–475. VLDB Endowment, 2003.

[Cormode *et al.*, 2004] Graham Cormode, Flip Korn, S. Muthukrishnan, and Divesh Srivastava. Diamond in the rough: finding hierarchical heavy hitters in multi-dimensional data. In *SIGMOD '04: Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, pages 155–166, New York, NY, USA, 2004. ACM.

[Cormode *et al.*, 2008] Graham Cormode, Flip Korn, S. Muthukrishnan, and Divesh Srivastava. Finding hierarchical heavy hitters in streaming data. *ACM Trans. Knowl. Discov. Data*, 1(4):1–48, 2008.

[Domingos and Pazzani, 1996] Pedro Domingos and Michael Pazzani. Beyond independence: Conditions for the optimality of the simple bayesian classifier. In *Machine Learning*, pages 105–112. Morgan Kaufmann, 1996.

[Frank and Asuncion, 2010] A. Frank and A. Asuncion. UCI machine learning repository, 2010.

[Gupta *et al.*, 2007] K. Gupta, B. Nath, and K. Ramamohanarao. Conditional random fields for intrusion detection. In *21st Intl. Conf. on Adv. Information Netw. and Appl.*, pages 203–208, 2007.

[Hastie *et al.*, 2003] T. Hastie, R. Tibshirani, and J. H. Friedman. *The Elements of Statistical Learning*. Springer, corrected edition, July 2003.

[Huang *et al.*, 2003] Jin Huang, Jingjing Lu, and Lambda Charles X. Ling. Comparing naive bayes, decision trees, and svm with auc and accuracy. In *in: Third IEEE International Conference on Data Mining, ICDM 2003*, pages 553–556. IEEE Computer Society, 2003.

[Lafferty *et al.*, 2001] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *Proc. 18th International Conf. on Machine Learning*, pages 282–289, 2001.

[Lohmann *et al.*, 2009] Daniel Lohmann, Wanja Hofer, Wolfgang Schröder-Preikschat, Jochen Streicher, and Olaf Spinczyk. CiAO: An aspect-oriented operating-system family for resource-constrained embedded systems. In *Proc. of USENIX ATEC*, Berkeley, USA, 2009. USENIX.

[Malouf, 2002] Robert Malouf. A comparison of algorithms for maximum entropy parameter estimation. In *COLING-02: proceedings of the 6th conference on Natural language learning*, pages 1–7, Morristown, NJ, USA, 2002. Association for Computational Linguistics.

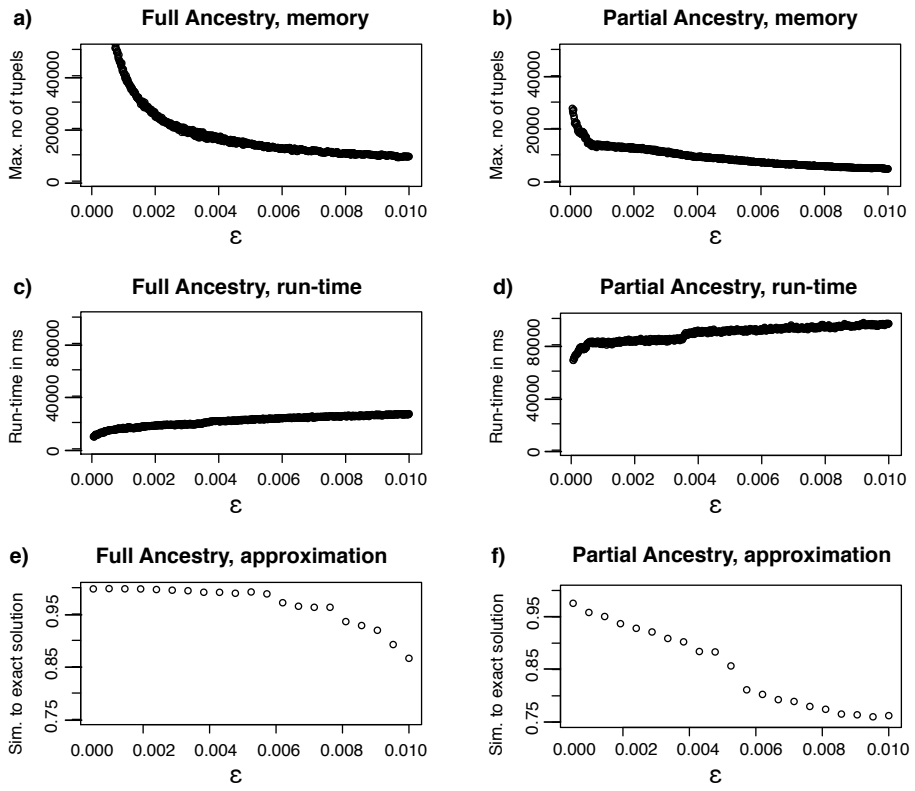


Figure 5: Memory consumption (a, b), run-time (c, d) and similarity to exact solution (e, f) of HHH algorithms (three-dimensional) with varying ε, $\phi = 0.001$ on biggest log file of application Rhythmbox.

[Nocedal, 1980] Jorge Nocedal. Updating quasi-newton matrices with limited storage. *Mathematics of Computation*, 35(151):773–782, 1980.

[Rabiner, 1989] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, Feb 1989.

[Schraudolph and Graepel, 2002] Nicol N. Schraudolph and Thore Graepel. Conjugate directions for stochastic gradient descent. In *ICANN '02: Proceedings of the International Conference on Artificial Neural Networks*, pages 1351–1358, London, UK, 2002. Springer-Verlag.

[Schraudolph *et al.*, 2007] Nicol N. Schraudolph, Jin Yu, and Simon Günter. A stochastic quasi-Newton method for online convex optimization. In Marina Meila and Xiaotong Shen, editors, *Proc. 11th Intl. Conf. Artificial Intelligence and Statistics (AIstats)*, volume 2 of *Workshop and Conference Proceedings*, pages 436–443, San Juan, Puerto Rico, 2007.

[Sha and Pereira, 2003] Fei Sha and Fernando Pereira. Shallow parsing with conditional random fields. In *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 134–141, Morristown, NJ, USA, 2003. Association for Computational Linguistics.

[Silberschatz *et al.*, 2010] Abraham Silberschatz, Peter Baer Galvin, and Greg Gagne. *Operating System Concepts*. Wiley Publishing, 2010.

[Sutton and McCallum, 2007] C. Sutton and A. McCallum. An Introduction to Conditional Random Fields for Relational Learning. In Lise Getoor and Ben Taskar, editors, *Introduction to Statistical Relational Learning*. MIT Press, 2007.

[Tartler *et al.*,] Reinhard Tartler, Daniel Lohmann, Wolfgang Schröder-Preikschat, and Olaf Spinczyk. Dynamic AspectC++: Generic advice at any time. In *The 8th Int. Conf. on Software Methodologies, Tools and Techniques*, Prague. IOS Press. (to appear).

[Tian *et al.*, 2007] S. Tian, S. Mu, and C. Yin. Sequence-similarity kernels for SVMs to detect anomalies in system calls. *Neurocomput.*, 70(4–6):859–866, 2007.

[Timm *et al.*, 2010] C. Timm, A. Gelenberg, F. Weichert, and P. Marwedel. Reducing the Energy Consumption of Embedded Systems by Integrating General Purpose GPUs. Technical Report 829, Technische Universität Dortmund, Fakultät für Informatik, 2010.

[Vishwanathan *et al.*, 2006] S. V. N. Vishwanathan, Nicol N. Schraudolph, Mark W. Schmidt, and Kevin P. Murphy. Accelerated training of conditional random fields with stochastic gradient methods. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*, pages 969–976, New York, NY, USA, 2006. ACM.

Workflow Analysis using Graph Kernels

Natalja Friesen and **Stefan Rüping**
Fraunhofer IAIS
Schloss Birlinghoven, St. Augustin, Germany
{natalja.friesen,stefan.rueping}@iais.fraunhofer.de

Abstract

Workflow enacting systems are a popular technology in business and e-science alike to flexibly define and enact complex data processing tasks. Since the construction of a workflow for a specific task can become quite complex, efforts are currently underway to increase the re-use of workflows through the implementation of specialized workflow repositories. While existing methods to exploit the knowledge in these repositories usually consider workflows as an atomic entity, our work is based on the fact that workflows can naturally be viewed as graphs. Hence, in this paper we investigate the use of graph kernels for the problems of workflow discovery, workflow recommendation, and workflow pattern extraction, paying special attention on the typical situation of few labeled and many unlabeled workflows. To empirically demonstrate the feasibility of our approach we investigate a dataset of bioinformatics workflows retrieved from the website myexperiment.org.[1]

1 Introduction

Workflow enacting systems are a popular technology in business and e-science alike to flexibly define and enact complex data processing tasks. A workflow is basically a description of the order in which a set of services have to be called with which input in order to solve a given task. Since the construction of a workflow for a specific task can become quite complex, efforts are currently underway to increase the re-use of workflows through the implementation of specialized workflow repositories.

Driven by specific applications, a large collection of workflow systems have been prototyped such as Taverna [Oinn *et al.*, 2004] or Triana [Taylor *et al.*, 2006].

As the high numbers of workflows can be generated and stored relatively easily it becomes increasingly hard to keep an overview about the available workflows. Workflow repositories and websites such as `myexperiment.org` tackle this problem by offering the research community the possibility to publish and exchange complete workflows. An even higher amount of integration has been described in the idea of developing a Virtual Research Environment (VRE, [Fraser, 2005]).

Due to the complexity of managing a large repository of workflows, data mining approaches are needed to support the user in making good use of the knowledge that is encoded in these workflows. In order to improve the flexibility of a workflow system, a number of data mining tasks can be defined:

Workflow recommendation: Compute a ranking of the available workflow with respect to their interestingness to the user for a given task. As it is hard to formally model the user's task and his interest in a workflow, one can also define the task of finding a measure of similarities on workflows. Given a (partial) workflow for the task the user is interested in, the most similar workflows are then recommended to the user.

Metadata extraction: Given a workflow (and possibly partial metadata), infer the metadata that describes the workflow best. As most approaches for searching and organizing workflow are based on descriptive metadata, this task can be seen as the automatization of the extraction of workflow semantics.

Pattern extraction: Given a set of workflows, extract a set of sub-patterns that are characteristic for these workflow. A practical purpose of these patterns is to serve as building block for new workflows. In particular, given several sets of workflows, one can also define the task of extracting the most discriminative patterns, i.e. patterns that are characteristic for one group but not the others.

Several approaches to these problems exists in the data mining literature, in particular on machine learning for recommender systems [Goderis, 2008]. However, existing methods usually consider workflows as an atomic entity, using workflow meta data such as its usage history, textual descriptions (in particular tags), or user-generated quality labels as descriptive attributes. While these approaches can deliver high quality results, they are limited by the fact that all these attributes require either a high user effort to describe the workflow (to use text mining techniques), or a frequent use of each workflow by many different users (to mine for correlations).

Figure 1 shows the evolution of information about a workflow over its lifetime. In the construction phase, mainly technical information about the workflow itself is generated. Possibly starting from a workflow pattern, several versions of a workflow are generated and tested. In the submission phase, the workflow is made public for possible re-use. Here, the owner gives a more or less detailed description about the purpose of the workflow and about the task that it is supposed to achieve. In the final phase, the re-use of the workflow by different users begins, generating information about workflow usage, such as correlations

[1] This paper also appears at the Workshop on Service-Oriented Knowledge Discovery (SoKD 2010)

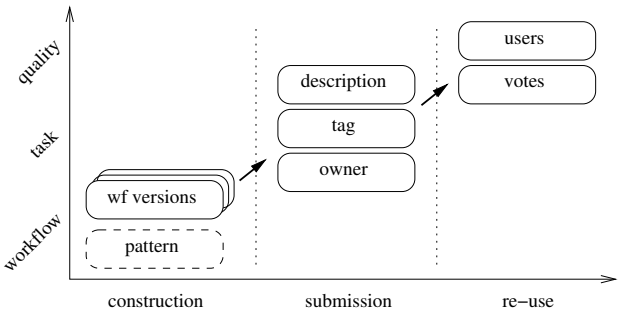


Figure 1: Workflow data: over the lifetime of a workflow (horizontal axis), information on different levels of generality is generated (vertical axis)

with usage of other workflows. User possibly vote on the quality of the workflow.

In this paper we are interested in supporting the user in constructing the workflow and reducing the manual effort of workflow tagging. The reason for the focus on the early phases of workflow construction is that in practice it can be observed that often users are reluctant to put too much effort into describing a workflow; they are usually only interested in using the workflow system as a means to get their work done. A second aspect to be considered is that without proper means to discover existing workflows for re-use, it will be hard to receive enough usage information on a new workflow to start up a correlation-based recommendation in the first place.

To address these problems, we have opted to investigate solutions to the previously described data mining tasks that can be applied in the common situation of many unlabeled workflows, using only the workflow description itself and no meta data. Our work is based on the fact that workflows can be viewed as graphs. We will demonstrate that by the use of graph kernels it is possible to effectively extract workflow semantics and use this knowledge for the problems of workflow recommendation and metadata extraction. The purpose of this paper is to answer the following questions:

Q1: How good are graph kernels at performing the tasks of workflow recommendation without explicit user input? We will present an approach that is based on exploiting workflow similarity.

Q2: Can appropriate meta data about a workflow be extracted from the workflow itself? What can we infer about the semantics of a workflow and its key characteristics? In particular, we will investigate the task of tagging a workflow with a set of user-defined keywords.

Q3: How good does graph mining perform at a descriptive approach of workflow analysis, namely the extraction of meaningful graph patterns?

The remainder of the paper is structured as follows: Next, we will discuss related work in the area of workflow systems. In Section 3, we give a detailed discussion of representation of workflows and the associated metadata. Section 4 will present the approach of using graph kernels for workflow analysis. The approach will be evaluated on 4 distinct learning tasks on a dataset of bioinformatics workflows retrieved from the website `http://myexperiment.org` in Section 5. Section 6 concludes.

2 Related Work

Since workflow systems are getting more complicated, the development of effective discovery techniques particularly for this field has been addressed by many researcher during the last years. Public repositories that enables sharing of workflows are widely used both in business and scientific communities. While first steps toward supporting the user have been made, there is still a need to improve effectiveness of discovery methods and support the user in navigating in the space of available workflows. A detailed overview of different approaches for workflow discovery is given by Goderis [Goderis, 2008].

Most approaches are based on simple search functionalities and consider workflow as an atomic entity. Searching over workflow annotation like titles, textual description or discovery on the basis of user profiles belongs to basic capabilities of such academic repositories as myExperiment [Roure, 2009], BioWep [2], Kepler [3] or commercial systems like Infosense and Pipeline Pilot.

In [Goderis *et al.*, 2009] a detailed study about current practices in workflow sharing, re-using and retrieval is presented. To summarize, the need to take into account structural properties of workflows in the retrieval process was underlined by several users. The authors demonstrate that existing techniques are not sufficient and there is still a need for effective discovery tools. In [Goderis *et al.*, 2006] retrieval techniques and methods for ranking discovered workflows based on graph-subisomorphism matching are presented. Coralles [Corrales *et al.*, 2006] proposes a method for calculating the structural similarity of two BPEL (Business Process Execution Language) workflows represented by graphs. It is based on error correcting graph subisomorphism detection.

Apart from workflow sharing and retrieval, the design of new workflows is an immense challenge to users of workflow systems. It is both time-consuming and error-prone, as there is a great diversity of choices regarding services, parameters, and their interconnections. It requires the researcher to have specific knowledge in both his research area and in the use of the workflow system. Consequently, it is preferable for a researcher to not start from scratch, but to receive assistance in the creation of a new workflow.

A good way to implement this assistance is to reuse or repurpose existing workflows or workflow patterns (i.e. more generic fragments of workflows). An example of workflow re-use is given in [Goderis, 2008], where a workflow to identify genes involved in tolerance to Trypanosomiasis in East African cattle was reused successfully by another scientist to identify the biological pathways implicated in the ability of mice to expel the Trichuris Muris parasite.

In [Goderis *et al.*, 2005] it is argued that designing new workflows by reusing and re-purposing previous workflows or workflows patterns has the following advantages:

- Reduction of workflow authoring time
- Improved quality through shared workflow development
- Improved experimental provenance through reuse of established and validated workflows
- Avoidance of workflow redundancy

[2] http://bioinformatics.istge.it/biowep/

[3] https://kepler-project.org/

While there has been some research comparing workflow patterns in a number of commercially available workflow management systems [Van Der Aalst *et al.*, 2003] or identifying patterns that describe the behavior of business processes [White, March 2004], to the best of our knowledge there exists no work to automatically extract patterns. A pattern mining method for business workflows based on calculation of support values is presented in [Thom *et al.*, 2007]. However, the set of patterns that was used was derived manually based on an extensive literature study.

3 Workflows

A workflow is a way to formalize and structure complex data analysis experiments. Scientific workflows can be described as a sequence of computation steps together with predefined input and output that arise in scientific problem-solving. Such definition of workflow enables sharing analysis knowledge within scientific communities in convenient way. Since we want to provide the user with suggestions about possibly interesting workflows, we need to be able to compare workflows available in existing VREs with the one that the user has. That implies that workflows need a consistent representation.

3.1 Example: myExperiment.org

We consider the discovery of similar workflow in the context of a specific VRE called myExperiment [Roure *et al.*, 2008], MyExperiment has been developed to support sharing of scientific objects associated with an experiment. It is a collaborative environment where scientists can safely publish their workflows and experiment plans, share them with groups and find those of others. MyExperiment is designed to make it easy for scientists to contribute to a pool of scientific workflows, build communities and form relationships. Its goal is to enable scientists to share and reuse workflows, reduce the time-to-experiment, and avoid reinvention.

Figure 2 shows a workflow view in myExperiment. Each stored workflow is created by a specific user, is associated with a workflow graph, and contains metadata and certain statistics such as the number of downloads or rating. We split all available information about a workflow into four different groups: the workflow graph, textual data, user information, and workflow statistics. Next we will characterize each group in more detail. Graph properties of workflow are detailed described in 4

3.2 Textual Data

Each workflow in myExperiment has a title and a description text and contains information about the creator and date of creation. Furthermore, the associated tags annotate workflow by several keywords that facilitate searching for workflows and provide more precise results. Additionally, for workflows in Taverna format myExperiment displays visual previews, detailed information about single workflow components, such as type of inputs and outputs, number and type of certain processors.

3.3 User Information

MyExperiment was created not only as an environment for sharing workflows, but also as a social infrastructure for the researchers. The social component is realized by registration of users and allowing them to create profiles with different kind of personal information, details about their work and professional life. The members of myExperiment can form complex relationships with other members, such as creating or joining user groups or giving credit to others. All this information can be used in order to find the groups of users having similar research interests or working in related projects. In the end, this type of information can be used to generate the well known correlation-based recommendations of the type "users who liked this workflow also liked the following workflows...".

3.4 Workflow Statistics

As statistic data we consider information that is changing with the time, such as the number of views or downloads or the average rating. Statistic data can be very useful for providing a user with a workflow he is likely to be interested in. As we do not have direct information about user preferences, some of the statistics data, e.g. number of downloads or rating, can be considered as a kind of quality measure.

4 A Graph Mining Approach to Workflow Analysis

The characterization of a workflow by metadata alone is challenging because neither of these features give an insight into the underlying sub-structures of the workflow. It is clear that users do not always create a new workflow from scratch, but most likely re-use old components and sub-workflows. Hence, knowledge of sub-structures is important information to characterize a workflow completely.

The common approach to represent objects for a learning problem is to describe them as vectors in a feature space. However, when we handle objects that have important substructures, such as workflows, the design of a suitable feature space is not trivial. For this reason, we opt to follow a graph mining approach.

4.1 Frequent Subgraphs

Frequent subgraph discovery has received intense and still growing attention, since it has a wide range of applications areas. Frequently occurring subgraphs in a large set of graphs could represent important motifs in the data. Given a set of graphs $\mathcal{G}$, the support of a graph G ($S(G)$) is defined as the fraction of graphs in $\mathcal{G}$ in which G occurs. The problem of finding frequent patterns is defined as follows:

Given a set of graphs $\mathcal{G}$ and minimum support S_{min}. We want to find all connected subgraphs G that occur frequently enough (i.e. $S(G) >= S_{min}$) over the entire set of graphs. The output of the discovery process may contain a large number of such patterns.

4.2 Graph Kernels

Graph kernels, as originally proposed by [Gaertner *et al.*, 2003; Kashima and Koyanagi, 2002] provide a general framework for handling graph data structures by kernel methods. Different approaches for defining graph kernels exist.

A popular representation of graphs that is used for examples in protein modeling and drug screening are kernels based on cyclic patterns [Horváth *et al.*, 2004]. However, these are not applicable to workflow data, as workflows are by definition acyclic (because an edge between services A and B represents the relation "A must finish before B can start").

To adequately represent the decomposition of workflows into functional substructures, we apply the following approach: the set of graphs is searched for substructures (subtrees) that occur in at least a given percentage (support)

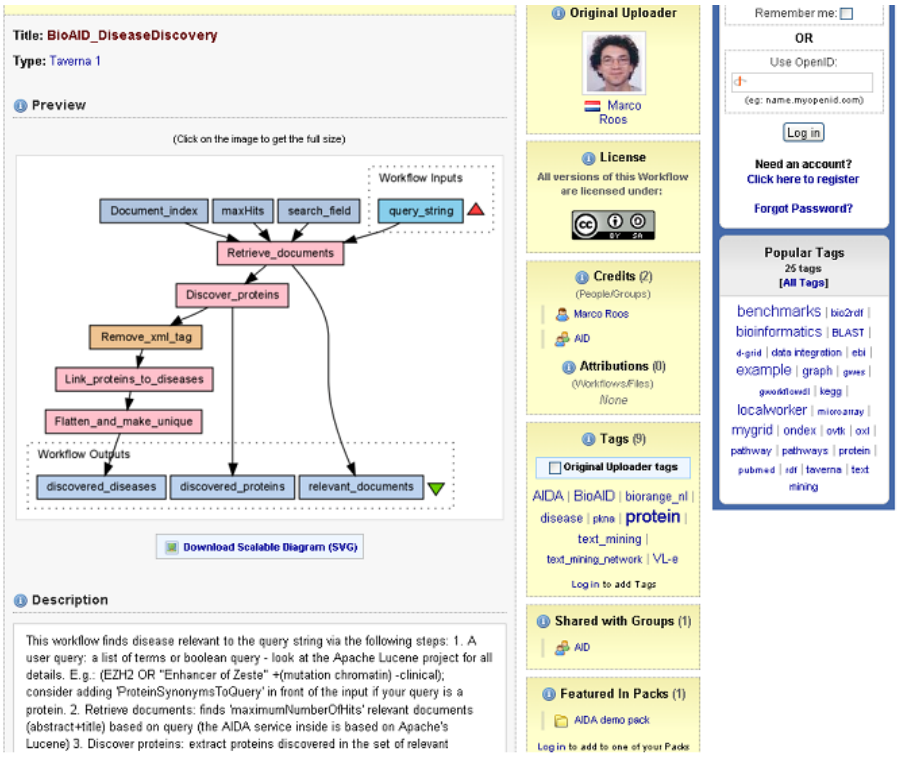


Figure 2: MyExperiment: workflow with associated metadata.

of all graphs. Then, the feature vector is composed of the weighted counts of the substructures. The substructures are sequences of labeled vertices that were produced by graph traversal. The length of a substructure is equal to the number of vertices in it. This family of kernels is called Label Sequence Kernels. The main difference among the kernels lies in how graphs are traversed and how weights are involved in computing a kernel. According to the extracted substructures, these are kernels based on walks, trees or cycles. In our work we used the kernels based on random walks with exponential weights proposed by Gärtner et al. [Gaertner *et al.*, 2003]. Since workflows are directed acyclic graphs, in our special case the hardness results of [3] (such random walks can not be enumerated in general case) no longer hold and we actually can enumerate all walks. This allows us to explicitly generate the feature space representation of the kernels by defining the attribute values for every substructure (walk). For each substructure s in the set of graphs, let k be the length of the substructure. Then, the attribute λ_s is defined as:

$$\lambda_s = \frac{\beta^k}{k!} \quad (1)$$

(and thus generate the explicit finite set of features directly).

if the graph contains the substructure s and $\lambda_s = 0$ else. Here β is a parameter that we have optimized by cross-validation.

The graph kernel between graphs G_1 and G_2 is now defined as

$$k(G_1, G_2) = \sum_{s \in sub(G_1) \cap sub(G_2)} \lambda_s$$

where the sum runs over all substructures s that are present in both G_1 and G_2. Note that if for a finite set of d substructures $s_1, \ldots, s_d$ we define

$$\lambda(G) = (\sqrt{\lambda_{s_1}}\mathbf{1}_{\{s_1 \in G\}}, \ldots, \sqrt{\lambda_{s_d}}\mathbf{1}_{\{s_d \in G\}})$$

this is equivalent to

$$k(G_1, G_2) = \lambda(G_1)^T \lambda(G_2).$$

A very important advantage of graph kernels approach for the discovery task is that distinct substructures can provide an insight into the specific behavior of the workflow.

4.3 Graph Representation of Workflows

A workflow can be formalized as a directed acyclic labeled graph. The workflow graph has two kind of nodes: regular nodes representing the computation operations and nodes defining input/output data structure. A set of edges shows information and control flow between the nodes. More formally, a workflow graph can be defined as a tuple $W = (N, T)$, where:

$N = \{C, I, O\}$

C = finite set of computation operations,

I/O = finite set of inputs or outputs

$T \subseteq N \times N$ = finite set of transitions defining control flow.

Labeled graphs contain an additional source of information. There are several alternatives to obtain node labels. On the one hand, users often annotate single workflow components by a combination of words or abbreviations. On the other hand, each component within workflow system has a signature and an identifier associated with it, e.g. in web-service WSDL format. User created labels suffer from subjectivity and diversity, e.g. the same node representing the same computational operation can be labeled in very different way. The first alternative again assumes some type of user input, so we opt to use the second alternative.

Figure 3 shows an example of such transformation obtained for a Taverna workflow [Oinn *et al.*, 2004]. While the left pictures shows a user annotated components the

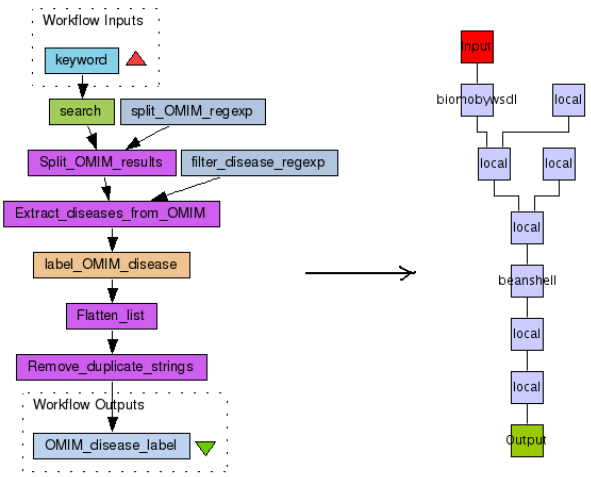


Figure 3: Transformation of Taverna workflow to the workflow graph.

right picture presents workflow graph on the next abstraction level.

5 Evaluation

In this section we illustrate the use of workflow structure and graph kernels in particular for workflow discovery and pattern extraction. We evaluate results on a real-world dataset of Taverna workflows. However, the same approach can be applied to other workflow systems, as long as we can obtain meaningful consistent labels from workflows.

5.1 Dataset

For the purposes of this evaluation we used a corpus of 300 real-world bioinformatics workflows retrieved from myExperiment [Roure *et al.*, 2008]. We chose to restrict ourselves to workflows that were created in Taverna workbench [Oinn *et al.*, 2004] in order to simplify the formatting of workflows. Since the application area of myExperiment is restricted to bioinformatics, we may be sure that there are sets of similar workflows there. The user feedback about similarity of workflow pairs is missing. Hence, we used semantic information to obtain workflows similarity. We made the assumption that workflows targeting the same tasks are similar. Under this assumption we used the cosine similarity of the vector of tags assigned to the workflow as a proxy for the true similarity. An optimization over the number of clusters resulted in 5 groups shown in Table 1. These tags indeed impose a clear structuring with few overlaps on the workflows.

5.2 Workflow Recommendation

In this section, we address Question **Q1**: How good are graph kernels at performing the tasks of workflow recommendation without explicit user input? The goal is to retrieve workflows that are "close enough" to a user's context. Therefore, we need to be able to compare workflows available in existing VREs with the user's one. As similarity measure we use graph kernel, which is equivalent to the cosine distance between the feature vectors consisting of the weighted counts of the frequent subgraphs.

We compare our approach based on graph kernels to several techniques representing the current state of the art [Goderis *et al.*, 2006]: matching of workflow graphs based on the size of the maximal common subgraph (MCS) and a method that considers a workflow as a bag of services. In addition to these techniques we also consider a standard text mining approach, whose main idea is that workflows are documents in XML format. The similarity of a workflow pair is then calculated as the cosine distance between the respective word vectors.

In our experiment we predict if two workflows belong to the same cluster. Table 2 summarizes the average performances of a leave-one-out evaluation for the four approaches. It can be seen that graph kernels clearly outperform all other approaches in accuracy and recall. For precision, MCS performs best, however, at the cost of a minimal recall. The recall of graph kernels ranks second and is close to the value of MCS.

Therefore, we conclude that graph kernels are suitable for the task of workflow recommendation based only on graph structure without explicit user input.

5.3 Workflow Tagging

We are now interested in Question **Q2** of extraction of appropriate metadata from workflows. As a prototypical piece of metadata, we investigate user-defined tags.

20 tags were selected that occur in at least 3% of all workflows. We use tags as proxies that represent the real-world task that a workflow can perform. Selected tags present 20 different classification problems. For each tag we would like to predict if it describes an given workflow. To do that we utilize graph kernels representation of workflow. We tested two algorithms: SVM and k-Nearest Neighbor. Table 3 shows results of tags prediction evaluated by 2-fold cross validation over 20 keywords. It can be seen that an SVM with graph kernels can predict the selected tags with high AUC and precision, while a Nearest Neighbor approach using graph kernels to define the distance achieves a higher recall.

We can conclude that the graph representation of workflow contains enough information to predict appropriate metadata.

5.4 Pattern extraction

Finally, we investigate question **Q3**, which deals with the more descriptive task of extracting meaningful patterns from sets of workflows that are helpful in the construction of new workflows.

We address the issue of extracting patterns that are particularly important within a group of similar workflows in several steps. First, we use a linear SVM to build a classification model based on the graph kernels. This model identifies all workflows which belong to the same group against workflows from other groups. Then we search for features having high weight value which the model considers as important. We performed such pattern extraction targeting consequently each workflow group. A 10-fold accuracy shows that this classification can be achieved with high accuracy, values ranging between 81.3% and 94.7%, depending on the class. However, we are more interested in the most significant patterns, which we determine based on the weight that was assigned by the SVM (taking different standard deviation into account). Examples of such patterns are described below.

Figure 4 shows an example of workflow patterns and the same pattern inside of one of workflows that it occurs in. It was considered as important for classifying workflow from

Group	Size	Most frequent tags	Description
1	30%	localworker, example, mygrid	Workflows using local scripts.
2	29%	bioinformatics, sequence, protein, BLAST, alignment, similarity, structure, search, retrieval	Sequence similarity search using the BLAST algorithm
3	24%	benchmarks	Benchmarks WFs.
4	6.7%	AIDA , BioAID, text mining, bioassist, demo, biorange	Text mining on biomedical texts using the AIDA toolbox and BioAID web services
5	6.3%	Pathway, microarray, kegg	Molecular pathway analysis using the Kyoto Encyclopedia of Genes and Genomes (KEGG)

Table 1: Characterization of workflow groups derived by clustering.

Method	Accuracy	Precision	Recall
Graph Kernels	$\mathbf{81.2} \pm 10.0$	71.9 ± 22.0	$\mathbf{38.3} \pm 21.1$
MCS	73.9 ± 9.3	$\mathbf{73.5} \pm 24.7$	4.8 ± 27.4
Bags of services	73.5 ± 10.3	15.5 ± 20.6	3.4 ± 30.1
Text Mining	77.8 ± 8.31	67.2 ± 21.5	31.2 ± 25.8

Table 2: Performance of workflow discovery.

group 2, which consists of workflows using the BLAST algorithm to calculate sequences similarity. The presented pattern is a sequence of components that are needed to run a BLAST service.

This example shows that graph kernels can be used to extract useful patterns. These patterns then can be recommended to the user during creation of a new workflow.

6 Conclusions

Workflow enacting systems have become a popular tool for the easy orchestration of complex data processing tasks. However, the design and management of workflows are a complex tasks. Machine learning techniques have the potential to significantly simplify this work for the user.

In this paper, we have discussed the usage of graph kernels for the analysis of workflow data. We argue that graph kernels are a good tool for the analysis of workflow data in the practical important situation where no meta data is available. This is due to the fact that the graph kernel approach allows to take decompositions of the workflow into its important substructures into account while allowing an flexible integration of these information contained into these substructures into several learning algorithms.

We have evaluated the use of graph kernels in the fields of workflow similarity prediction, metadata extraction, and pattern extraction. A comparison of graph-based workflow analysis with metadata-based workflow analysis in the field of workflow quality modeling showed that metadata-based approaches outperform graph-based approaches in this application. However, it is important to recognize that the goal of the graph-based approach is not to replace the metadata-based approaches, but to serve as an extension when no or few metadata is available.

References

[Corrales *et al.*, 2006] Juan Carlos Corrales, Daniela Grigori, and Mokrane Bouzeghoub. Bpel processes matchmaking for service discovery. In *In Proc. CoopIS 2006, Lecture Notes in Computer Science 4275*, pages 237–254. Springer, 2006.

[Fraser, 2005] M. Fraser. *Virtual Research Environments: Overview and Activity*. Ariadne, 2005.

[Gaertner *et al.*, 2003] Thomas Gaertner, Peter Flach, and Stefan Wrobel. On graph kernels: Hardness results and efficient alternatives. In *Proceedings of the 16th Annual Conference on Computational Learning Theory and 7th Kernel Workshop*, pages 129–143. Springer-Verlag, August 2003.

[Goderis *et al.*, 2005] Antoon Goderis, Ulrike Sattler, Phillip Lord, and Carole Goble. Seven bottlenecks to workflow reuse and repurposing. *The Semantic Web ISWC 2005*, pages 323–337, 2005.

[Goderis *et al.*, 2006] Antoon Goderis, Peter Li, and Carole Goble. Workflow discovery: the problem, a case study from e-science and a graph-based solution. In *ICWS '06: Proceedings of the IEEE International Conference on Web Services*, pages 312–319. IEEE Computer Society, 2006.

[Goderis *et al.*, 2009] Antoon Goderis, Paul Fisher, Andrew Gibson, Franck Tanoh, Katy Wolstencroft, David De Roure, and Carole Goble. Benchmarking workflow discovery: a case study from bioinformatics. *Concurr. Comput. : Pract. Exper.*, (16):2052–2069, 2009.

[Goderis, 2008] Antoon Goderis. *Workflow re-use and discovery in bioinformatics*. PhD thesis, School of Computer Science, The University of Manchester, 2008.

[Horváth *et al.*, 2004] Tamás Horváth, Thomas Gärtner, and Stefan Wrobel. Cyclic pattern kernels for predictive graph mining. In *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 158–167, New York, NY, USA, 2004. ACM.

[Kashima and Koyanagi, 2002] Hisashi Kashima and Teruo Koyanagi. Kernels for semi-structured data. In *ICML '02: Proceedings of the Nineteenth International Conference on Machine Learning*, pages 291–298, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc.

Method	AUC	Precision	Recall
Nearest Neighbors	0.54 ± 0.18	0.51 ± 0.21	$\mathbf{0.58} \pm 0.19$
SVM	$\mathbf{0.85} \pm 0.10$	$\mathbf{0.84} \pm 0.24$	0.38 ± 0.29

Table 3: Workflows tagging based on graph kernels.

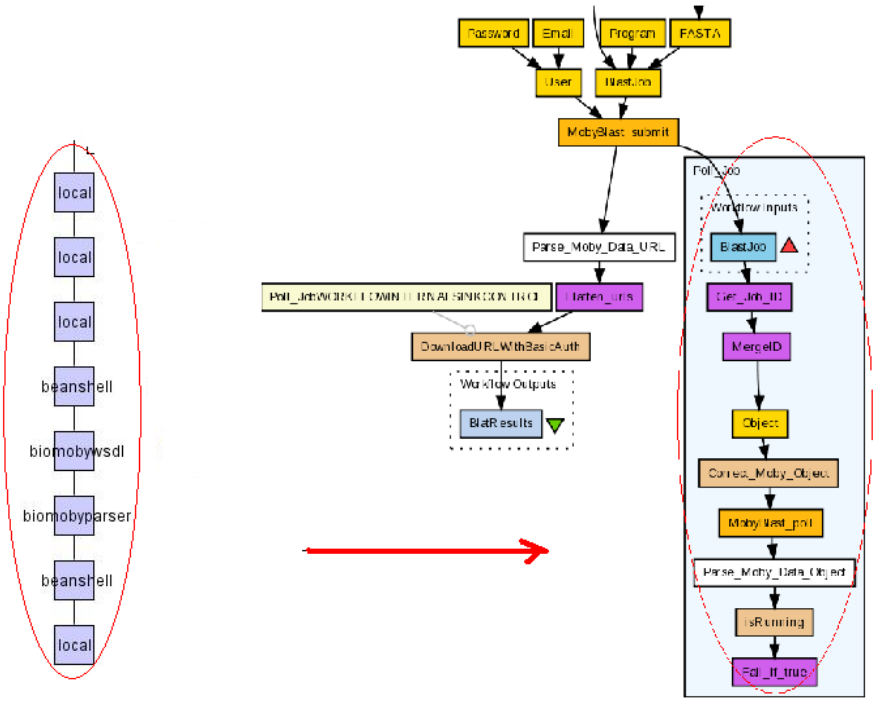


Figure 4: Example of workflow graph.

[Oinn *et al.*, 2004] T Oinn, M.J. Addis, J. Ferris, D.J. Marvin, M. Senger, T. Carver, M. Greenwood, K Glover, M. R. Pocock, A. Wipat, and P. Li. Taverna: a tool for the composition and enactment of bioinformatics workflows. *Bioinformatics*, 20(17):3045–3054, June 2004.

[Roure *et al.*, 2008] David De Roure, Carole Goble, Jiten Bhagat, Don Cruickshank, Antoon Goderis, Danius Michaelides, and David Newman. myexperiment: Defining the social virtual research environment. In *4th IEEE International Conference on e-Science*, pages 182–189. IEEE Press, December 2008.

[Roure, 2009] Robert Stevens David De Roure. The design and realisation of the myexperiment virtual research environment for social sharing of workflows. 2009.

[Taylor *et al.*, 2006] Ian J. Taylor, Ewa Deelman, Dennis B. Gannon, and Matthew Shields. *Workflows for e-Science: Scientific Workflows for Grids*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[Thom *et al.*, 2007] Lucineia Thom, Cirano Iochpe, and Manfred Reichert. Workflow patterns for business process modeling. In *Proc. of the CAiSE'06 Workshops - 8th Int'l Workshop on Business Process Modeling, Development, and Support (BPMDS'07)*, page Vol. 1. Trondheim, Norway, 2007.

[Van Der Aalst *et al.*, 2003] W. M. P. Van Der Aalst, A. H. M. Ter Hofstede, B. Kiepuszewski, and A. P. Barros. Workflow patterns. *Distrib. Parallel Databases*, 14(1):5–51, 2003.

[White, March 2004] Stephen A. White. Business process trends. In *Business Process Trends*, March, 2004.

Visually summarizing the Evolution of Documents under a Social Tag (Resubmission)

André Gohr
Leibniz Institute of Plant Biochemistry, Halle, Germany

Myra Spiliopoulou
Otto-von-Guericke University, Magdeburg, Germany

Alexander Hinneburg
Martin-Luther University Halle-Wittenberg, Germany

Abstract

Tags are intensively used in social platforms to annotate resources: Tagging is a social phenomenon, because users do not only annotate to organize their resources but also to associate semantics to resources contributed by third parties. This leads often to semantic ambiguities: Popular tags are associated with very disparate meanings, even to the extend that some tags (e.g. "beautiful" or "toread") are irrelevant to the semantics of the resources they annotate. We propose a method that learns a topic model for documents under a tag and visualizes the different meanings associated with the tag.

Our approach deals with the following problems. First, tag miscellany is a temporal phenomenon: tags acquire multiple semantics gradually, as users apply them to disparate documents. Hence, our method must capture and visualize the evolution of the topics in a stream of documents. Second, the meanings associated to a tag must be presented in a human-understandable way; This concerns both the choice of words and the visualization of all meanings. Our method uses AdaptivePLSA, a variation of Probabilistic Latent Semantic Analysis for streams, to learn and adapt topics on a stream of documents annotated with a specific tag. We propose a visualization technique called Topic Table to visualize document prototypes derived from topics and their evolution over time. We show by a case study how our method captures the evolution of tags selected as frequent and ambiguous, and visualizes their semantics in a comprehensible way. Additionally, we show the effectiveness by adding alien resources under a tag. Our approach indeed visualizes hints to the added documents.

The full article is published in the proceedings of the International Conference on Knowledge Discovery and Information Retrieval, KDIR 2010 and the SciTePress Digital Library. A copy of the PDF-file is available from `http://users.informatik.uni-halle.de/~hinnebur/PS_Files/2010KDIR_TT.pdf`.

One Clustering Process Fits All - A Visually Guided Ensemble Approach

Martin Hahmann, Dirk Habich, Maik Thiele, Wolfgang Lehner
Dresden University of Technology
Database Technology Group
01187, Dresden, Germany
dbinfo@mail.inf.tu-dresden.de

Abstract

Looking back on the past decade of research on clustering algorithms, we witness two major and apparent trends: 1) The already vast amount of existing clustering algorithms, is continuously broadened and 2) clustering algorithms in general, are becoming more and more adapted to specific application domains with very particular assumptions. As a result, algorithms have grown complicated and/or very scenario-dependent, which made clustering a hardly accessible domain for non-expert users. This is an especially critical development, since, due to increasing data gathering, the need for analysis techniques like clustering emerges in many application domains. In this paper, we oppose the current focus on specialization, by proposing our vision of a usable, guided and universally applicable clustering process. In detail, we are going to describe our already conducted work and present our future research directions.

1 Introduction

To obtain an optimal clustering, knowledge in the domain of clustering algorithms and the domain of the application data is essential. In practise, the average application users, e.g, biologists, are usually experts of the data domain but only have limited knowledge about the available tools for clustering. Therefore, clustering is a challenging task for this domain experts [Jain and Law, 2005]. The reasons can be briefly summarized as follows: (i) The selection of a clustering algorithm is critical, since, in general, only a fraction of the available algorithms is known to the user. Moreover, most algorithms are tailored to specific tasks and are thus, not appropriate for every data set. (ii) Another obstacle is parameterization, which offers many degrees of freedom but provides nearly no support. (iii) Finally, the interpretation of results, is complicated by the multitude of existing visualization and validation techniques, which are also not universally applicable.

In our opinion, the nearly unlimited options and the high degree specialization of algorithms are the major obstacles, which prevent non-expert users from the successful application of clustering algorithms. Therefore, we state that users first of all need a universally applicable process, rather than a zoo of highly customized clustering algorithm. Based on the paradigm of ensemble-clustering, we want to abandon specialization and develop a unifying clustering process. This process integrates the user, offers guidance and allows the purposeful navigation through the available clustering solutions, as well as the step-by-step construction of a satisfying clustering result, by adjustment of the ensemble clustering and on-demand generation of additional clusterings for parts of the dataset. To realize such an unifed clustering process, research effort has to be done on three areas: algorithms (Section 2), usability (Section 3) and architecture. In the remainder of this paper we will present our research results we obtained so far [Hahmann *et al.*, 2009; 2010b; 2010a] and state the open challenges in the respective research areas (Section 4). We omit the architecture area and refer to the following papers [Habich *et al.*, 2007a; 2007b; 2010]. Finally, we conclude the paper with a brief summary in Section 5

2 Algorithm Area - The Clustering Process

In this section, we introduce the underlying algorithmic plaform for our unifed clustering process—*Flexible Clustering Aggregation (FCA)* [Hahmann *et al.*, 2009]. The basic concept of FCA is clustering aggregation, which combines different clusterings of a dataset into one result to increase quality and robustness [Hahmann *et al.*, 2009; Gionis *et al.*, 2007]. Different aggregation approaches are known, where the pairwise assignment approach is considered as the most capable one [Boulis and Ostendorf, 2004; Caruana *et al.*, 2006; Topchy *et al.*, 2004; Zeng *et al.*, 2002; Dimitriadou *et al.*, 2001; Dudoit and Fridlyand, 2003; Fred, 2001; Fred and Jain, 2003; Frossyniotis *et al.*, 2002; Gionis *et al.*, 2007; Habich *et al.*, 2006]. This approach evaluates each object pair of a dataset, determining whether it is assigned (i) to the same cluster or (ii) to different clusters. The aggregate is constructed by selecting the most frequent of these two pairwise assignments for each object pair and setting it in the result clustering. All existing aggregation techniques lack controllability, thus an aggregation result can only be adjusted through modification and re-computation of the input clusterings.

Our *Flexible Clustering Aggregation (FCA)* [Hahmann *et al.*, 2009] tackles this issue. The key approach of our technique is to change the aggregation input from hard to soft clusterings [Bezdek, 1981]. These assign to each object its relative degree of similarity with all clusters instead of a hard assignment to just one cluster. Such assignments can be (i)generated by specific algorithms like *fuzzy c-means* [Bezdek, 1981] or (ii) calculated from arbitrary clustering results, using refinement techniques like *a-posteriori* [Zeng *et al.*, 2002]. Up to now, we only utilize *fuzzy c-means* to generate the clusterings for our ensembles.

In a soft clustering result, each datapoint $x_i|(1 \leq i \leq n)$

of a dataset $\mathcal{D}$ is assigned to all k clusters $c_j|(1 \leq j \leq k)$ of a clustering C to a certain degree. Thus, the assignment information of x_i in C is denoted as a vector v_i with the components $v_{ip}(1 \leq p \leq k)|0 < v_{ip} < 1 and \sum_{p=1}^{k} v_{ip} = 1$ describing the relation between x_i and the p-th cluster of C. This fine-grained information allows, e.g., the identification of undecidable cluster assignments given when objects have identical maximal similarities with multiple clusters. Assume a clustering with $k = 3$, and an object x_i with ${v_i}^{\top} = (0.4, 0.4, 0.2)$. Using this assignment, we cannot decide whether x_i belongs to c_1 or c_2, although c_3 can be excluded. Based on this, it is easy to see that the worst case regarding decidability is given for assignments with $\forall v_{ip}(1 \leq p \leq k) = 1/k$, since they do not even allow the exclusion of clusters when it comes to clear cluster affiliations. Of these two kinds of undecidable assignments, we name the first *balanced* and the second *fully balanced* [Hahmann *et al.*, 2009].

To incorporate this additional information, we expanded the pairwise assignment cases for the aggregation by adding an undecidable case that is valid for object pairs containing undecidable assignments. Furthermore, we derived a significance measure for pairwise assignments on that basis. This measure incorporates the intra-pair similarity of soft assignments and their decidability. The lower bound for decidability is defined as 0 or as an impossible decision and is given for the mentioned undecidable cluster assignments. The upper bound of 1 is given for objects with a single degree of similarity v_{ip} approaching 1 while all others approach 0. Basically, decidability shows the distance of v_i to the *fully balanced* assignment.

With this significance score, pairwise assignments are filtered and classified as undecidable if they do not exceed a certain significance threshold. Aggregation control or result adjustment, respectively, is exercised by this filtering and the handling of undecidable pairwise assignments during aggregation. Since *undecidable* is no valid option for a final object assignment, two handling strategies exist: one assumes that undecidable pairs are part of the same cluster, while the other assumes the opposite. These strategies and the filtering threshold act as parameters, allowing the merging or splitting of clusters without modifying the input clusterings[Hahmann *et al.*, 2009].

Generally, the relation between parameters and the clustering result is one of cause and effect. Parameters like k for k-means or ε for DBSCAN *cause* different *effects* in the clustering result, e.g., the fusion of clusters or changes in their size. To achieve a certain effect, it is crucial to know its associated cause, which is quite challenging. The FCA method overcomes this by allowing the direct specification of desired *effects*, namely: *merge* for fewer clusters or *split* for more clusters. In our original work [Hahmann *et al.*, 2009], these *effects* could only be applied to the whole clustering and were thus mutually exclusive. In our recent work, we enhanced the algorithm so that those *effects* can be applied to individual clusters.

3 Usablity Area - Visual Decision Support

Until now, merging and splitting have been mutually exclusive and had to be set for the whole clustering. This is suffcient if the bulk of clusters requires the same operation, but it effectively prevents an individual handling of clusters. In tight coupling with our ensemble approach, our efforts in the useability area shall enable users to interpret the obtained clustering result and assist them in the decision on whether or not clusters are stable and should be merged or spilt. With this, the result quality can be iteratively refined, whereas the provided support keeps the iteration count low.

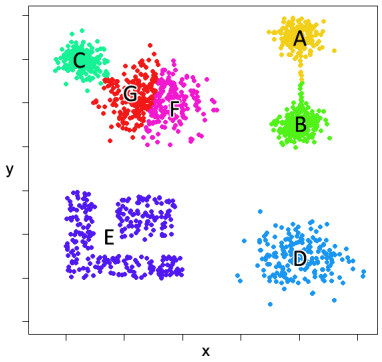


Figure 1: Example aggregate.

To efficiently support result interpretation and adjustments, we developed a visualization concept that is tightly coupled to our aggregation method [Hahmann *et al.*, 2010b]. In general, two major groups of data/clustering visualizations can be distinguished: The first one is data-driven and tries to depict all objects and dimensions of a dataset, which leads to incomprehensible presentations for datasets exceeding a certain scale. The second one is result-driven and thus relatively scale-invariant. For example, a clustering can be depicted as a bar chart showing relative cluster sizes and/or additional values like mean or standard deviation. While the first group often shows too much information, the second one often shows not enough. So, we positioned our approach as a hybrid between those groups, by visualizing the result and the relations between data and result, which are already incorporated in the soft input of our aggregation. In compliance with Shneiderman's mantra, 'overview first, zoom and filter, then details-on-demand' [Shneiderman, 1996], our visualization features three interactive views: overview, cluster composition and relations (c&r), and the attribute view.

With this, we want to enable the user to determine the clusters that need no adjustment and to decide which ones should be merged or split with our aggregation algorithm. We define stable clusters according to the general objective of clustering, that asks for clusters with high internal similarity that are well separated from each other. Clusters that not fullfill these criteria are candidates for adjustment. To illustrate our approach, the clustering aggregate depicted in Figure 1 is used as our paper example. It has been generated using our ensemble clustering and shows a partitioning result that needs some adjustments. In all subsequent figures, clusters are identified via color.

3.1 Overview

The overview is the first view presented to the user and depicted in Figure 2. This view is completely result-driven, i.e., only characteristics of the clustering aggregate are shown. The dominant circle represents the clusters of the aggregate, whereas each circle segment corresponds to a cluster whose percental size correlates with the segment's size. The radar-like gauge located on the left shows the distances between the prototypes (centroids) of all clusters. The mapping between centroids in the radar and circle segment is done via color. The radar shows a distance graph, where vertices represent centroids, and edges—invisible in our visualization—represent the Eu-

clidean distance between centroids in the full dimensional data space. Therefore, the radar is applicable for high-dimensional data. Since all our views are basically result-driven, we can also handle high-volume datasets without problems. The overview provides the user with a visual summary of the clustering result, allowing a first evaluation of the number of clusters and relations between clusters expressed by distance and size.

3.2 Cluster Composition and Relations

If the user identifies clusters of interest in the overview, e.g., two very close clusters like the pink (F) and red (G) ones in Figure 1, they can be selected individually to get more information about them, thus performing *'zoom and filter'*. Cluster selection is done by rotation of the main circle. As soon as a cluster is selected, the composition and relations (c&r) view depicted in Figure 3 (for cluster F) is displayed. The selected cluster's composition is shown by the row of histograms on the right. All histograms feature the interval $[0, 1]$ with ten bins of equal width. From the left to the right, they show the distribution of: (i) fuzzy assignment values, (ii) significance scores for all object-centroid pairs, and (iii) significance scores for all object-object pairs in the selected cluster. For details concerning these scores, refer to [Hahmann *et al.*, 2009]. Certain histogram signatures indicate certain cluster states, e.g., a stable and compact cluster is given if all three histograms show a unimodal distribution with the mode—ideally containing all objects—situated in the right-most (highest significance) bin.

Let us regard the signature of the example depicted in Figure 3. The histograms show that many of the object-centroid and pairwise assignments are not very strong. This indicates that there are other clusters (G in the example) that strongly influence the selected cluster objects, which leaves the chance that these clusters could be merged. To support such assumptions, the relations between clusters have to be analyzed. For this, the two 'pie-chart' gauges and arcs inside the main circle are used. The smaller gauge shows the degree of 'self-assignment' of the selected cluster, while the other one displays the degree of 'shared assignment' and its distribution among the remaining clusters. These degrees are calculated as follows: each fuzzy object assignment is a vector with a sum of 1, consisting of components ranged between 0 and 1, indicating the relative degree of assignment to a certain cluster, i.e., each vector-dimension corresponds to a cluster. The degree of self-assignment is calculated by summing up all components in the dimension corresponding to the selected cluster. This sum is then normalized and multiplied with 100 to get a percental score. The shared assignment is generated in the same fashion for each remaining cluster/dimension. The target and strength of relations between the selected cluster and others is described by the color and size of the shared-assignment slices. For easy identification, the displayed arcs show these cluster-to-cluster relations by connecting clusters, where the stroke width shows the strength of the relation.

If a cluster is not influenced by others, it shows a very high degree of self-assignment with no outstanding relations to other clusters. In contrast, the example in Figure 3 shows that the selected cluster has a noticeable relation to the red cluster. This supports the merge assumption and furthermore indicates which other cluster should be part of a possible merge. To get additional information, the inter-cluster distances can be analyzed. For this, the user can employ the 'radar', showing that both clusters in our example are relatively close to each other (the selected cluster is encircled), or switch on additional distance indicators (*'details-on-demand'*), as shown in Figure 4. These display the ratio of centroid-to-centroid distances—like the radar—and minimum object-to-object distances between the selected and the remaining clusters. If this ratio approaches 1, the respective clusters are well separated and the colored bars are distant. In our example, this is the case for all clusters except for the red one, where both bars nearly touch each other, showing that the minimal object distance between the clusters is much smaller than the centroid distance. With this, the user can now savely state that the pink and the red cluster should be merged. To double-check, the red cluster can be selected and should show similar relations to the pink one.

With the c&r view, it is also possible to evaluate whether or not a cluster should be split. Candidates for a split show the following: In all three histograms, the mode of the distribution is located in one of the medium-significance bins. Additionally, they feature a reduced degree of self-assignment, but in contrast to the merge case, they have equally strong relations to the remaining clusters and are well separated in terms of the radar and distance indicators. Unfortunately, these characteristics are no clear indication for a split, e.g., non-spherical clusters can exhibit the same properties. To gain more certainty in decisions for split candidates, the attribute view has been developed.

3.3 Attribute View

When we look at attributes in terms of clustering, we can state the following: If an attribute has a uniform or unimodal distribution (in the following Φ), it is not useful for clustering because the objects of the dataset cannot be clearly separated in this dimension. In contrast, bi- or multi-modal distributions are desired, since they can be used for object separation. When we look at attributes on the cluster level, this is inverted. Regarding a cluster, it is desirable that all of its attributes have unimodal distributions, since this shows high intra-cluster homogeneity. A multimodal-distributed attribute would imply that the cluster could be further separated in this dimension. Generally, we desire the following: On the dataset level, attributes should be dissimilar to Φ, while on the cluster level, they should resemble it as closely as possible. These are the basics for our attribute view.

To calculate the similarity to Φ, we use a straightforward approach. We generate histograms, on the dataset and cluster level, for each attribute. From the histogram bins, those that are local maxima are selected. From each maximum, we iterate over the neighboring bins. If a neigboring bin contains a smaller or equal number of objects, it is counted and the next bin is examined; otherwise, the examination stops. With this, we can determine the maximum number of objects and bins of this attribute that can be fitted under Φ. This is the value we display in the attribute view. In Figure 5, the attribute view is depicted for the violet cluster E from our example. There are two hemispheres and a band of numbers between them. The band shows the attributes of the dataset, ordered by our computed values, and is used to select an attribute for examination (selection has a darker color). The small hemisphere on the right shows the global behavior of attributes. Each curve represents an attribute, while for the selected attribute, the area under its

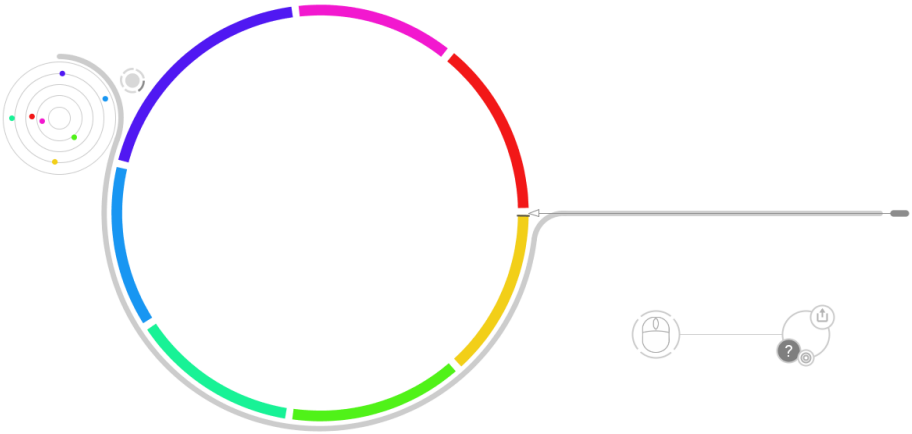


Figure 2: AUGUR overview showing clusters and inter-cluster distances.

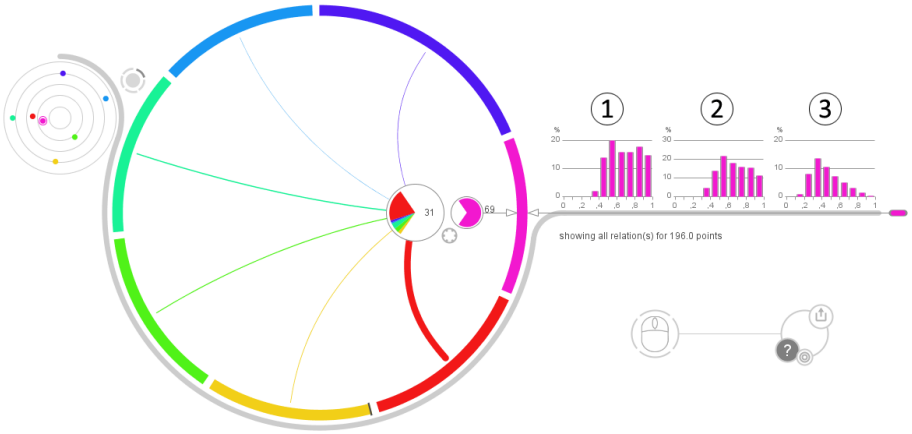


Figure 3: AUGUR c&r view showing composition and relations for the pink cluster.

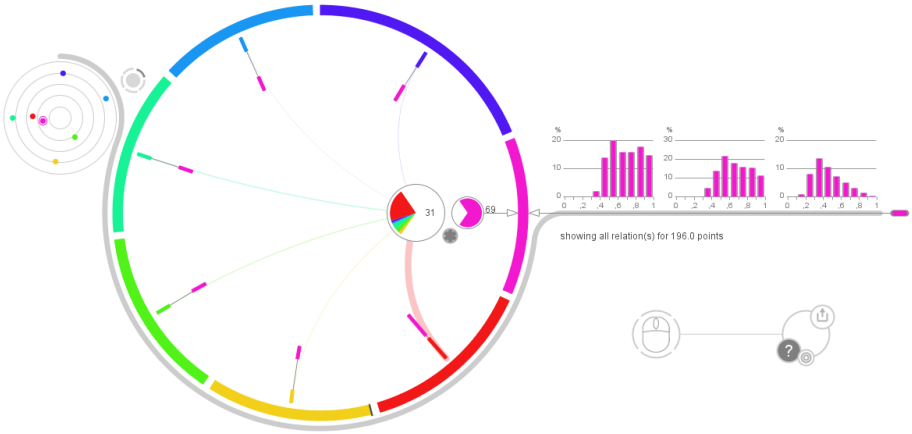


Figure 4: AUGUR c&r view with activated distance indicators.

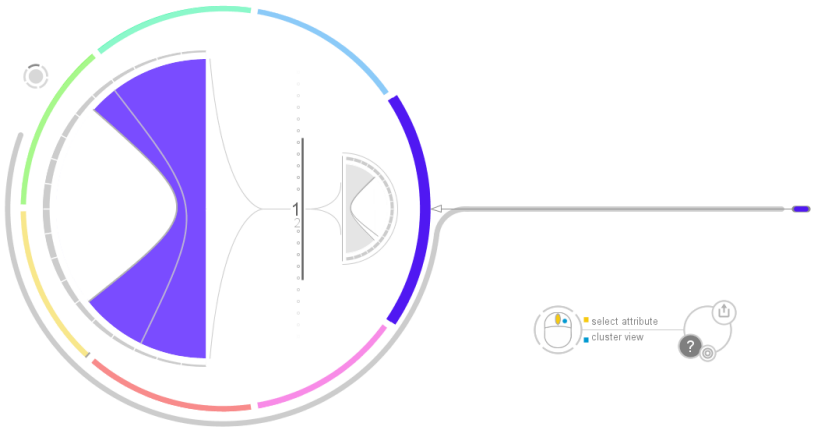


Figure 5: AUGUR attribute view indicating a split for the violet cluster.

curve is colored. The hemisphere itself consists of two 90-degree scales, the upper for the percentage of objects and the lower for the percentage of bins that can be fitted under Φ. The start and end point of each curve show the values for the attribute on these scales. If all objects and bins fit under Φ, a vertical line is drawn and there is no color in the hemisphere. All this also applies to the left hemisphere showing the attribute in the selected cluster. For our example in Figure 5, we selected attribute 1.

We can see a large colored area, showing that more than 50% of the objects and bins do not fit under Φ. If, in addition, the selected cluster shows split characteristics in the c&r view, the user may assume that this cluster should be split. The benefit of this view lies in the fast and easy interpretability. More color in the left hemisphere indicates a higher split possibility, while the amount of color in the right hemisphere acts as a measure of confidence for the left. In terms of Shneiderman's mantra, this view can either be considered as *'details-on-demand'* or as an *'overview'* and *'zoom and filter'* for the attribute space.

3.4 Feedback

The basic idea of our unifed clustering process is to take advantage of the tight coupling between our two components and change the focus for parameterization from the whole clustering to individual clusters. For this, we use the following workfow: At first, the visualization of an initial clustering aggregate like our running example is presented to the user. In this view, he/she evaluates all clusters and looks for those that need adjustment. For those, the effects merge or split would be identified as appropriate. The respective parameter is then passed to the aggregation algorithm and only the specified clusters are subjected to a new aggregation cycle, while the rest of the result is kept. In sucessive steps, the user adjusts the clustering using the provided feedback operations until the result is satisfying.

4 Future Work

Although, we already acquired several results and components for our unifying process, much work still needs to be done. In the algorithmic area the utilization of the additional information, stored in soft clusterings, has proven beneficial. Therefore we want to expand its employment in our unifying process. Our short term goal is the development of a soft density-based clustering algorithm and a soft hierarchical method later on. In combination with the soft partitioning algorithm we used so far (fuzzy c-means[Bezdek, 1981]), these three algorithms, will provide us with a good coverage, for the generation of our cluster ensemble. To keep parameterization easy, the aggregations role as an abstraction layer must be developed further, so that users are provided with a stable and algorithm-independent interface [Hahmann *et al.*, 2010b] for the adjustment of clusterings. For this it is necessary to specifically implement parameters like *merge* for each algorithm.

Besides parameterization, the whole area of usability will be developed further. The employment of density-based and hierarchical clustering algorithms, leads to different views of the data. Depending on the generating algorithm, different information can be derived from the obtained clusterings. This information must be examined, refined and communicated to the user in the form of novel visualization concepts. Thereby, the focus lies on convenient presentation metaphors, that present information, necessary for the user to navigate through our process without flooding him/her with too much input. In addition, to improve the guidance during navigation, we will implement a semi-automatic recommender system for the selection of appropriate parameters/feedback in arbitrary stages of our unifying process. Another major aspect of our future work in the area of usability will be the integration of novel and intuitive interaction platforms like, e.g., Apple's iPad or Nintendo's Nunchuk controllers. These platforms have already shown in practice, that intuitive interaction simplifies access to unfamiliar technologies.

Concerning the architecture domain, we further focus on scalability for all our used components. Regarding the interaction platform previously mentioned, we will also regard the question, if the components of our process can be distributed and where each component should be executed. An example setting could execute the complete algorithmic stack on high performance hardware, while process interaction is done on a portable device.

In summary, our long term goal, is to combine our existing components with the future work, outlined in this section to form our unifying clustering process. In its final state, this process will provide an initial clustering as starting point, which the user adjusts and refines in a step-by-step fashion. While the user navigates through the process, he/she is offered guidance for parameter and algorithm selection, as well as hints on which parts of the clustering still need refinement.

5 Conclusion

In our work we oppose the specialisation of clustering and claim, that one clustering process can fit all application scenarios. To realize this claim, we concentrate on controllable ensemble-clustering, guided user interaction and emerging computation architectures. We are fully aware, that our position is bold and controversial. Nevertheless, our recent work supports our overall vision of a unifed visually guided clustering process.

In this paper, we summarized our current status on the algorithmic and useability area. We focus on enabling the user to evaluate an ensemble clustering result and on providing decision support for result refinement with our extended aggregation algorithm proposed in [Hahmann *et al.*, 2009]. There already exist a multitude of cluster visualization techniques [Hinneburg, 2009], which mostly try to visualize all objects of the dataset and are thus limited if data sets exceed a certain size. Furthermore, some of these techniques use complex visual concepts, which can hinder interpretation. In contrast, our visualization is tightly coupled to our aggregation method [Hahmann *et al.*, 2009]. We do not try to visualize all objects of the data set but concentrate on the presentation of clusters as well as cluster-cluster and cluster-object relations, derived from soft cluster assignments. This result- and relation-oriented approach allows the interpretation of data sets with arbitrary volume/dimensionality and supports the user in making decisions concerning result refinement via the mentioned *split* and *merge* actions. In addition, focusing on *'what'* to visualize, namely clusters and relations, allows the use of well-known and simple visual elements, e.g., pie charts and histograms, when it comes to *'how'* to visualize.

References

[Bezdek, 1981] James C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. New York: Plenum, 1981.

[Boulis and Ostendorf, 2004] Constantinos Boulis and Mari Ostendorf. Combining multiple clustering systems. In *Proc. of PKDD*, 2004.

[Caruana *et al.*, 2006] Rich Caruana, Mohamed Farid Elhawary, Nam Nguyen, and Casey Smith. Meta clustering. In *Proc. of ICDM*, 2006.

[Dimitriadou *et al.*, 2001] Evgenia Dimitriadou, Andreas Weingessel, and Kurt Hornik. Voting-merging: An ensemble method for clustering. In *Proc. of ICANN*, 2001.

[Dudoit and Fridlyand, 2003] Sandrine Dudoit and Jane Fridlyand. Bagging to improve the accuracy of a clustering procedure. *Bioinformatics*, 19(9), Jun 2003.

[Fred and Jain, 2003] Ana L. N. Fred and Anil K. Jain. Robust data clustering. In *Proc. of CVPR*, 2003.

[Fred, 2001] Ana L. N. Fred. Finding consistent clusters in data partitions. In *Proc. of MCS*, 2001.

[Frossyniotis *et al.*, 2002] Dimitrios S. Frossyniotis, Minas Pertselakis, and Andreas Stafylopatis. A multi-clustering fusion algorithm. In *Proc. of SETN*, 2002.

[Gionis *et al.*, 2007] Aristides Gionis, Heikki Mannila, and Panayiotis Tsaparas. Clustering aggregation. *TKDD*, 1(1), 2007.

[Habich *et al.*, 2006] Dirk Habich, Thomas Wächter, Wolfgang Lehner, and Christian Pilarsky. Two-phase clustering strategy for gene expression data sets. In *Proc. of SAC*, 2006.

[Habich *et al.*, 2007a] Dirk Habich, Steffen Preissler, Wolfgang Lehner, Sebastian Richly, Uwe Aßmann, Mike Grasselt, and Albert Maier. Data-grey-boxweb services in data-centric environments. In *Proceedings of the 2007 IEEE International Conference on Web Services (ICWS 2007, July 9-13, Salt Lake City, Utah, USA)*, pages 976–983, 2007.

[Habich *et al.*, 2007b] Dirk Habich, Sebastian Richly, Mike Grasselt, Steffen Preißler, Wolfgang Lehner, and Albert Maier. Bpel^{DT} - data-aware extension of bpel to support data-intensive service applications. In *Proceedings of the 2nd Workshop of Emerging Web Services Technology in conjunction with the 5th IEEE European Conference on Web Services (WEWST 2007, November 26, Halle/Salle, Germany)*, 2007.

[Habich *et al.*, 2010] Dirk Habich, Wolfgang Lehner, Sebastian Richly, and Wolfgang Lehner. Using cloud technologies to optimize data-intensive service applications. In *Proceedings of the 3rd International Conference on Cloud Computing (CLOUD 2010, Miami, FL, USA, June 5-10)*, 2010.

[Hahmann *et al.*, 2009] Martin Hahmann, Peter Volk, Frank Rosenthal, Dirk Habich, and Wolfgang Lehner. How to control clustering results? flexible clustering aggregation. In *Advances in Intelligent Data Analysis VIII*, pages 59–70, 2009.

[Hahmann *et al.*, 2010a] Martin Hahmann, Dirk Habich, and Wolfgang Lehner. V2i: A model process for accessible clustering. In *Appears as Demo at the 16th ACM SIKDD Conference on Knowledge Discovery and Data Mining (KDD 2010, Washington, DC, USA, July 25-28)*, 2010.

[Hahmann *et al.*, 2010b] Martin Hahmann, Dirk Habich, and Wolfgang Lehner. Visual decision support for ensemble clustering. In *Proceedings of the 22nd International Conference in Scientific and Statistical Database Management (SSDBM 2010, Heidelberg, Germany, June 30 - July 2)*, pages 279–287, 2010.

[Hinneburg, 2009] Alexander Hinneburg. Visualizing clustering results. In *Encyclopedia of Database Systems*, pages 3417–3425. 2009.

[Jain and Law, 2005] Anil Jain and Martin Law. Data clustering: A users dilemma. *Pattern Recognition and Machine Intelligence*, pages 1–10, 2005.

[Shneiderman, 1996] Ben Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *VL '96: Proceedings of the 1996 IEEE Symposium on Visual Languages*, page 336, Washington, DC, USA, 1996. IEEE Computer Society.

[Topchy *et al.*, 2004] Alexander P. Topchy, Behrouz Minaei-Bidgoli, Anil K. Jain, and William F. Punch. Adaptive clustering ensembles. In *Proc. of ICPR*, 2004.

[Zeng *et al.*, 2002] Yujing Zeng, Jianshan Tang, Javier Garcia-Frias, and Guang R. Gao. An adaptive meta-clustering approach: Combining the information from different clustering results. In *Proc. of CSB*, 2002.

Efficient frequent connected subgraph mining in graphs of bounded tree-width*

Tamás Horváth[1,2] **Jan Ramon**[3]

[1]Dept. of Computer Science III, University of Bonn, Germany
[2]Fraunhofer IAIS, D-53754 Sankt Augustin, Germany
[3]Department of Computer Science, Katholieke Universiteit Leuven, Belgium
tamas.horvath@iais.fraunhofer.de jan.ramon@cs.kuleuven.be

Abstract

The frequent connected subgraph mining problem, i.e., the problem of listing all connected graphs that are subgraph isomorphic to at least a certain number of transaction graphs of a database, cannot be solved in output polynomial time in the general case. If, however, the transaction graphs are restricted to forests then the problem becomes tractable. In this paper we generalize the positive result on forests to graphs of bounded tree-width. In particular, we show that for this class of transaction graphs, frequent connected subgraphs can be listed in incremental polynomial time. Since subgraph isomorphism remains NP-complete for bounded tree-width graphs, the positive complexity result of this paper shows that efficient frequent pattern mining is possible even for computationally hard pattern matching operators.

*A long version of this extended abstract appeared in *Theoretical Computer Science* **411**(31-33), 2784–2797 , 2010.

On the Need of Graph Support for Developer Identification in Software Repositories

[position paper]

Aftab Iqbal and **Marcel Karnstedt**
Digital Enterprise Research Institute (DERI)
National University of Ireland, Galway, Ireland
firstname.lastname@deri.org

Abstract

Software repositories from open-source projects provide a rich source of information for a wide range of tasks. However, one issue to overcome in order to make this information useful is the accurate identification of developers. This is a particular challenge, as developers usually use different IDs in different repositories of one project, but usually there is no kind of dictionary or similar available to map the different IDs to real-world persons. Often, they even use different IDs in the same repository. We show that the few methods suggested so far are not always appropriate to overcome this problem. Further, we highlight related techniques from other areas and discuss how they can be applied in this context. We particularly focus on the idea of applying graph-based methods and argue for the benefits we expect from that.

1 Motivation

In *Software Engineering*, many tools with underlying repositories have been introduced to support the collaboration of distributed software development. Research has shown that these software repositories contain rich amount of information about software projects. By mining the information contained in these software repositories, practitioners can depend less on their experience and more on the historical data [Hassan, 2008]. However, software repositories are commonly used only as record-keeping repositories and rarely for design decision processes [Diehl *et al.*, 2009]. The Mining Software Repositories (MSR) field analyzes the rich information available in these software repositories to discover interesting facts about the software projects [Diehl *et al.*, 2009]. Examples of software repositories are [Hassan *et al.*, 2005] :

1. **source control repositories** store changes to the source code as development progresses;
2. **bug repositories** keep track of the software defects;
3. **archived communications** between project developers record rationale for decisions throughout the life of a project.

Software developers use these repositories to interact with each other or to solve software-related problems. For example, source-code and bugs are quite often discussed on bug tracking systems or project mailing lists. By extracting rich information from these repositories, one can guide decision processes in modern software development. For example, data in a source control repository could be analyzed to extract the authorship information, which could be linked to additional authorship information extracted from author tags of source-code files. This could allow to keep track of which source files were committed by a developer in different periods of time. With "Linked Data Driven Software Development" (LD2SD) [Iqbal *et al.*, 2009], we have introduced a Linked Data-based methodology to relate data across software repositories explicitly and unambiguously. The so created interlinked data sets can be used for querying and browsing the related information that exists in these software repositories.

An excerpt of an exemplary RDF representation of Java source-code using our LD2SD approach is shown in listing 1. Further, an example of RDF representation of a SVN Commit is shown in listing 2.

```
1 @prefix baetle:
      <http://baetle.googlecode.com/svn/ns/#> .
2 @prefix ld2sd: <http://ld2sd.deri.org/LD2SD/ns#> .
3 @prefix : <http://ld2sd.deri.org/data/Java/> .
4  :connect a baetle:JavaClass;
5 baetle:author
      <http://ld2sd.deri.org/data/author/Developer_A>
      ;
6 ld2sd:imports "java.io.IOException" ,
7        "javax.servlet.ServletException" ;
8 ld2sd:hasMethod :connect#getConnection .
```

Listing 1: An exemplary Java RDFication.

```
1 @prefix baetle:
      <http://baetle.googlecode.com/svn/ns/#> .
2 @prefix owl: <http://www.w3.org/2002/07/owl#> .
3 @prefix : <http://ld2sd.deri.org/data/Svn2RDF/> .
4  :275 a baetle:Committing ;
5 baetle:modified
      <http://svn.deri.org/trunk/org/link/connect.java>
      ;
6 baetle:author
      <http://ld2sd.deri.org/data/author/Dev_A> .
7 <http://svn.deri.org/trunk/org/link/connect.java> a
8 baetle:JavaSource ;
9 owl:sameAs <http://ld2sd.deri.org/data/Java/connect>
      .
```

Listing 2: An exemplary Subversion RDFication.

The listings indicate one major problem in the context of mining these repositories: often developers use different identities for each software repository and sometimes multiple identities for the same repository, while interacting with these software repositories in different context. Using distinct identities for different repositories makes developers appear as different entities. Hence we need methods and techniques to correctly link the different identities of software developers. This is a main requirement for, among others, being able to keep track of the developers' activities in different software repositories.

There exist only few works discussing this particular issue (see Section 3). However, these works either lack in

details or propose simplified solutions. In a series of experiments, we found that the so far proposed methods are not sufficient for all cases. We reflect on these results in Section 4. Thus, we argue that we need more sophisticated identification methods. Specifically, we propose to use graph-based mechanisms. This is based on the observation that all the repository data can be represented as graphs. Moreover, these graphs are related and similar to each other, as they are based on related and similar activities from the same set of developers. Finally, we conclude and propose to use these graph-based and related approaches in Section 5.

2 Identities in Software Repositories

In order to interact with the many software repositories that are part of an open-source project, developers usually require to adopt an identity for each repository. Often, developers use multiple identities for the same repository [Robles and Gonzalez-Barahona, 2005]. Different types of identities that developers use in software repositories discussed by Robles et al. are (also summarized in Table 1):

1. In a source-code file, developers appear with many different identities, such as real life names, email addresses, SVN identifiers and sometimes the combination of real life name and email addresses;
2. Developers usually use multiple email addresses to send mails to the project mailing list. Sometimes the email headers contain the *<name, email>* pair, which helps to link the email address to the developer;
3. To commit source-code on the source control repository, developers use a separate account on the versioning system;
4. Bug tracking systems require an account associated with an email address.

Data Source	Identities
Source Code	Name Surname
Source Code	username@domain.com
Source Code	Name[username@domain.com]
Source Code	$subversionID
Mailing List	username@domain.com
Mailing List	Name Surname
Versioning System	$subversionID
Bug Tracking	username@domain.com

Table 1: Identities found in different software repositories [Robles and Gonzalez-Barahona, 2005].

In open-source projects, project outsiders (i.e., contributors) also submit source-code patches to the project mailing list. They cannot contribute code directly to the source control repository, because they are not invited to be part of the core group of developers of that project [Hassan, 2008]. Such contributors quite often mention their names in the javadoc[1] of the source-code files while modifying them to fix a particular bug or adding a new feature request to the project. When core developers finally commit the changes of contributors to the source-control repository, they very often mention the name of the contributor who provided the patch in the summary of the commit (e.g., "*Patch provided by DeveloperABC ...*"). Such information is very useful for analysis if extracted properly and interlinked to the correct developer name.

However, the fact that the sets of users differ between the repositories increases the difficulties of mapping IDs between them. This is a further challenge beside the fact that we have no 1:1 mapping, neither in one repository nor over different repositories. Actually, we often encounter an n:m mapping, e.g., several email addresses might belong to the same developer and multiple developers might use the same email address. An example of the later is an address like `developers@apache.org`. While this might be obvious in this case, it poses a significant challenge for automated approaches for developer identification.

[1] `http://java.sun.com/j2se/javadoc/writingdoccomments/`

3 Related Work

To the best of our knowledge, there are only a few published works on identifying and relating the different identities that developers use to interact with different tools in the field of software engineering. In [Bird *et al.*, 2006], Bird et al. proposed an approach to produce a list of *<name,email>* identifiers by parsing the emails and clustering them. The clustering algorithm to measure the similarity between every pair of IDs is based on string similarity between names, between emails, between names and emails, etc. Two IDs with a similarity measure lying below a pre-defined threshold value are placed into the same cluster. The authors use different approaches to compute the similarity measures between every pair of IDs, some of which we tested on our dataset to validate the effectiveness of these approaches (cf. Section 4.1 and Section 4.2). We found that their approach failed to provide satisfying results in our case. We use this as a motivation to argue for sophisticated approaches, such as graph-based methods.

[Robles and Gonzalez-Barahona, 2005] discusses the problem of developer identification in general, but the work lacks in details about the heuristics they propose to identify and match the different identities of developers. This makes it difficult to validate their approaches for solving this problem. The authors propose a technique to build one identity from another by extracting the "real life" name from email addresses, such as *nsurname@domain.com, name.surname@domain.com* etc. This is an approach based on pre-defined name schemes, a variant that we also evaluate for our case in Section 4.3. Similarly, this method did not provide satisfying results for the data set we investigated. Further, they try to match user names obtained from CVS to email addresses (excluding the domain after "@"). This approach relies on string similarity again.

In general, the problem is related to duplicate detection. While research in this area mostly refers to identifying duplicates in the same data set, the techniques might be mapped to the case of matching over different data sets. However, they are tailor-made for identifying different IDs of the same developer inside one repository. [Naumann and Herschel, 2010] provides a nice overview of this research direction. Interestingly, the authors also reflect on graph-based duplicate detection. The general idea is to use structural measures of graphs in order to identify different nodes that are similar to each other, wrt. these structural features. Two very similar nodes are likely referring to the same person. Bringing these approaches to the case of multiple graphs forms a basis for several works on network de-anonymisation [Narayanan and Shmatikov, 2009; Wondracek *et al.*, 2010]. The idea is similar: use the structural information from one (partially) known graph and use it to identify similar nodes in another graph, one of which we have only (maybe limited) structural information. Again, those nodes that are very similar are likely referring to the same person. This requires input graphs that are somehow related to each other. Further, the more adversary information we have, the higher accuracy we can achieve. We explore this approach further for our case, where we have several different input graphs, in Section 5.

4 Preliminary Results

In this section, we present results gained on data from a representative open-source project: *Apache Tomcat*[2]. We executed some experiments to evaluate the methods described in [Bird *et al.*, 2006] and [Robles and Gonzalez-Barahona, 2005]. In the following, we refer to *email addresses* by using only the term *email*. We gathered data by parsing the emails from *Apache Tomcat* mailing lists starting from 2005 to date. For every email containing a written name *and* the according email, we extracted the *from* and *to* field from the email header to produce a list of *<name,email>* pairs. We parsed 49,927 emails from the mailing list and produced 1261 distinct *<name,email>* pairs. We use these results as a ground proof to validate the different techniques that could be utilized to interlink the different identities of a developer. The reason for using data from a set where we actually know the exact matching is simple: it is the only way to provide some ground truth. It is not possible to extract similar information for other types of IDs. The only, clearly impractical, way would be to ask all involved persons for the different IDs they use. However, we strongly believe that the gained results are significant for these other types of IDs as well. Moreover, the methods we tested were proposed for matching email addresses.

To check the effectiveness of each approach, we first used the approach to find a matching developer name for each email. Then, we computed two versions of precision P and recall R values for each approach using the following equations:

$$P_1 = \frac{\#\text{emails with at least one correct match}}{\#\text{matched emails}} \quad (1)$$

$$R_1 = \frac{\#\text{emails with at least one correct match}}{\#\text{total emails}} \quad (2)$$

$$P_2 = \frac{\#\text{emails with exactly one correct match}}{\#\text{matched emails}} \quad (3)$$

$$R_2 = \frac{\#\text{emails with exactly one correct match}}{\#\text{total emails}} \quad (4)$$

By determining these different types we gain some interesting additional knowledge. For P_1 and R_1 we count a hit if we found at least one correct match, i.e., even if we found more incorrect matches. This provides an "optimistic" quality assessment, as the correct match(es) are found. Anyhow, in reality one would still have to filter out the wrong matches. Thus, we determine a "pessimistic" quality assessment in P_2 and R_2. If there exist correct *and* wrong matches for a given email this is *not* regarded as a hit.

In order to compute the similarities between developer names and email addresses or between two email addresses (see Section 4.1 and Section 4.2), we used the Levenshtein edit distance [Ukkonen, 1985] algorithm, as suggested in [Bird *et al.*, 2006]. The match(es) for an email are those developer names with the lowest distance. Clearly, there is no sense in allowing arbitray large distances, as this would mean that one string can be transformed into a completely other one. Thus, we tested three different threshold values:

1. the maximal length of both strings
2. the minimal length of both strings
3. a fixed threshold value of 4

[2] `http://tomcat.apache.org/`

4.1 Similarity between Names and Emails

In the first test we matched email addresses (excluding the domain after "@") and developer names by doing a pairwise comparison. Based on the above described approaches, we computed the precision and recall values, which are shown in Table 2.

Threshold	P_1 **in %**	R_1 **in %**
Maximum length	67.1	57.1
Minimum length	57.4	46.3
Fix Threshold	54.4	30.3
Threshold	P_2 **in %**	R_2 **in %**
Maximum length	24.3	20.1
Minimum length	18.4	15.3
Fix Threshold	48.4	27.2

Table 2: Names-email similarity results.

The results for P_1 and R_1 are actually quite good, where the maximal-length threshold seems to be the best choice. However, the accuracy is not high enough. Moreover, the values of P_2 and R_2 show that these methods produce a lot of false matches. As the fixed threshold seems to be the most "stable" in that context, we can learn that high thresholds tend to match very different strings – and thus create more false positives. In general, all methods are not very well suited for practice on a data set like ours.

4.2 Similarity between Emails

Besides names and emails, it is very likely that email addresses are textually similar to each other if they belong to the same developer. Thus, in the second test, we computed the pairwise similarities between email addresses (excluding the domain after "@") and determined match(es) for one email by choosing those with the lowest distance. This was also suggested in [Bird *et al.*, 2006]. The resulting precision and recall values are shown in Table 3.

Threshold	P_1 **in %**	R_1 **in %**
Maximum length	3.2	3.2
Minimum length	3.1	3.0
Fix Threshold	7.2	2.4
Threshold	P_2 **in %**	R_2 **in %**
Maximum length	2.0	2.0
Minimum length	1.5	1.5
Fix Threshold	3.5	1.3

Table 3: Email similarity results.

Clearly, this idea does not work at all for our data. The low values show that assuming similarity between emails in order to find matches is an absolutely inappropriate solution for our case.

4.3 Matching based on Name Schemes

More likely, email addresses are built from the "real life" name of a developer. As suggested by [Robles and Gonzalez-Barahona, 2005], in the third test we tried to identify matching developer names for emails (excluding the domain after "@") by checking the relation between them based on different name schemes. For example, `aftab.iqbal@deri.org` matches to `Aftab Iqbal` based on the name scheme *name.surname*. Based on our observations on different open-source projects we selected different name schemes, which developers quite often used to build their email addresses : *name.s* (e.g., *aftabi*), *n.surname* (e.g., *aiqbal*), *n.s* (e.g., *ai*), *na.su* (e.g., *afiq*), *name.surname* (e.g., *aftabiqbal*), *e.surname* (e.g., *biqbal*) and *name* (e.g., *aftab*). Note that the dot in a name scheme is only for illustration purposes – in the actual matching we do not use dots and ignore them in emails if present. Precision and recall as computed for each of the name scheme are shown in Table 4.

NameScheme	P_1 in %	R_1 in %
name.s	83.0	2.0
n.surname	96.0	9.2
n.s	33.3	1.0
na.su	50.0	0.2
name.surname	99.4	14.0
e.surname	100.0	0.6
name	72.0	3.5

NameScheme	P_2 in %	R_2 in %
name.s	65.5	2.0
n.surname	94.0	9.0
n.s	5.0	0.1
na.su	0.0	0.0
name.surname	99.4	14.0
e.surname	100.0	0.6
name	72.0	3.5

Table 4: Namescheme - email results.

To our surprise, most of the schemes result in very high precision values, for both types of quality measures. This means that if we find a match with these techniques, it is very likely a correct match. However, the in contrast very low recall values show that this method is capable of identifying only a handful of all matches. Thus, on its own, it is also not suited for our use case.

Summarizing, we can state that all three tested approaches do not prove to be suited for our case. As we have some good results in parts, it seems to be promising to combine these techniques with some more advanced approaches. By inspecting the precision and recall values, we can conclude that combining the methods among themselves is not promising. Thus, in the following section, we discuss graph-based methods to complement them.

5 Graph Support: A New Way of Developer Identification

As briefly mentioned in Section 3, there are several approaches that tackle a problem similar to the one discussed here on the basis of graphs. The main principle is that if we have two related graphs, we can use the structural features of nodes from one graph to reflect on the nodes from the other graph. This is particularly efficient in the context of social network graphs [Narayanan and Shmatikov, 2009; Wondracek *et al.*, 2010], i.e., graphs built from social relations between persons.

The approach suggested by us builds on the same principle. Out of the different repositories, we can construct several graphs: developer-to-file relations, developer-to-developer relations (e.g., working on the same files), email-to-email conversations, bug-to-developer relations, etc. While not all of these graphs are from social interactions, they are clearly all related to each other. Using some base knowledge, for instance gained from the name scheme methods described above, we can try to identify pairs of matching nodes. In order to map SVN IDs to developer names we could for instance build "file fingerprints" (similar to the group fingerprints in [Wondracek *et al.*, 2010]) and compare them. There are plenty of different options. The main task is to identify the kind of relations that connect two or more graphs.

Unfortunately, we are currently restricted in the graphs we can use. This is mainly due to the fact that we have ground truth only for emails and developer names. The only graphs we can extract from the *Apache Tomcat* project containing both types of IDs are email-to-email conversations and developer-to-developer relations (based on common files). However, we found that both graphs are similar in certain structural features, such as centrality values and degree distribution. This is quite intuitive, as developers that handle more classes (central position, higher degree) can be assumed to communicate more by email. In the very near future we plan to develop according methods for developer identification based on this observation. Later, we plan to extend these approaches to all types of graphs and IDs we find in the open-source software repositories. Moreover, we believe it is promising to extend the raw graphs we find by graphs derivated from other knowledge. For example, we will analyse the actual content of email conversations. Using according NLP techniques, we will very likely find more relations between developers and other mail users.

With confidence, we could show that existing methods are not suited for all cases. With the experiences from duplicate detection and de-anonymization, we strongly believe to have identified the right way for sophisticated developer identification. We are looking forward to explore this very interesting and relevant field of research in more detail in very near future.

Acknowledgements

This material is based upon works jointly supported by the Science Foundation Ireland under Grant No. SFI/08/CE/I1380 (Lion-2) and under Grant No. 08/SRC/I1407 (Clique: Graph & Network Analysis Cluster)

References

[Bird *et al.*, 2006] Christian Bird, Alex Gourley, Prem Devanbu, Michael Gertz, and Anand Swaminathan. Mining email social networks. In *MSR '06: Int. Workshop on Mining Software Repositories*, pages 137–143, 2006.

[Diehl *et al.*, 2009] Stephan Diehl, Harald C. Gall, and Ahmed E. Hassan. Guest editor's introduction: Special issue on mining software repositories. *Empirical Softw. Eng.*, 14(3):257–261, 2009.

[Hassan *et al.*, 2005] Ahmed E. Hassan, Audris Mockus, Richard C. Holt, and Philip M. Johnson. Guest editor's introduction: Special issue on mining software repositories. *IEEE Trans. Softw. Eng.*, 31(6):426–428, 2005.

[Hassan, 2008] Ahmed E. Hassan. The Road Ahead for Mining Software Repositories. In *Future of Software Maintenance (FoSM) at Int. Conf. on Software Maintenance(ICSM)*, 2008.

[Iqbal *et al.*, 2009] A. Iqbal, O. Ureche, M. Hausenblas, and G. Tummarello. LD2SD: Linked Data Driven Software Development. In *Int. Conf. on Software Engineering and Knowledge Engineering (SEKE 09)*, 2009.

[Narayanan and Shmatikov, 2009] Arvind Narayanan and Vitaly Shmatikov. De-anonymizing social networks. In *SP '09: IEEE Symposium on Security and Privacy*, pages 173–187, 2009.

[Naumann and Herschel, 2010] Felix Naumann and Melanie Herschel. An introduction to duplicate detection. *Synthesis Lectures on Data Management*, 2(1):1–87, 2010.

[Robles and Gonzalez-Barahona, 2005] Gregorio Robles and Jesus M. Gonzalez-Barahona. Developer identification methods for integrated data from various sources. *SIGSOFT Softw. Eng. Notes*, 30(4):1–5, 2005.

[Ukkonen, 1985] Esko Ukkonen. Algorithms for approximate string matching. *Inf. Control*, 64(1-3):100–118, 1985.

[Wondracek *et al.*, 2010] Gilbert Wondracek, Thorsten Holz, Engin Kirda, and Christopher Kruegel. A practical attack to de-anonymize social network users. Technical report, 2010. `http://www.iseclab.org/papers/sonda-TR.pdf`.

Separate-and-conquer Regression

Frederik Janssen and Johannes Fürnkranz
TU Darmstadt, Knowledge Engineering Group
Hochschulstraße 10, D-64289 Darmstadt
{janssen,juffi}@ke.tu-darmstadt.de

Abstract

In this paper a rule learning algorithm for the prediction of a numerical target variable is presented. It is based on the separate-and-conquer strategy and the classification phase is done by a decision list. A new splitpoint generation method is introduced for the efficient handling of numerical attributes. It is shown that the algorithm performs comparable to other regression algorithms where some of them are based on rules and some are not. Additionally a novel heuristic for evaluating the trade-off between consistency and generality of regression rules is introduced. This heuristic features a parameter to directly trade off the rule's consistency and its generality. We present an optimal setting for this parameter based on an optimization on several data sets.

1 Introduction

The accurate prediction of a numerical target variable is an important task in machine learning. There are several domains that can benefit from regression methods. For example, in the domain of financial data, it is a crucial issue to predict the volume of a credit. Here, classification algorithms can only provide a decision of whether or not a credit should be given but are not capable of predicting its size.

In the machine learning community the main task still is to predict a categorical outcome but through the last years the task of regression has gained more and more interest. Regression has its roots in the statistical community from where several algorithms were proposed over the years. The list includes the popular linear regression that is very efficient but still shows a good performance. The main advantage in using means of machine learning lies in the comprehensibility of the models. For instance, simple IF-THEN rules are directly interpretable by a data miner. Rules and trees are the two variants of interpretable models used in machine learning. As rules are typically more expressive because they are able to overlap, the goal of this work is the design of a rule learning system that on the one hand has a performance that is comparable to state-of-the-art algorithms and that on the other hand yields models that are still human-readable.

There are several strategies to induce a set of rules. Some of them rely on the gradient-descent algorithm for finding a rule ensemble that optimizes some loss function. Others convert given trees into sets of rules. However, one of the most popular strategy in classification is the so-called separate-and-conquer paradigm. Due to its simplicity and its good performance in classification[1], we decided to use this strategy to design the algorithm.

The paper is started with a brief recapitulation of related work. It is continued by a short introduction of separate-and-conquer rule learning for classification. Then the adaptations that are necessary to extent separate-and-conquer rule learning for classification to regression are specified. Some error measures are introduced and the handling of numerical attributes is described. Then the experimental setup and the evaluation methods are specified and the method for optimizing the parameters of the algorithm is described. The following section describes the results and the last one concludes the paper.

2 Related work

The separate-and-conquer strategy is not used frequently for learning regression rules. Exceptions include *predictive clustering rules* (PCR) [Ženko *et al.*, 2005], the FRS system [Demšar, 1999], which is a reimplementation of the FORS system [Karalič and Bratko, 1997], and M5RULES [Holmes *et al.*, 1999; Quinlan, 1992; Wang and Witten, 1997] which generates the regression rules from model trees and uses linear models in the head of the rules. Predictive clustering rules are generated by modifying the search heuristic of CN2 [Clark and Niblett, 1989]. Instead of *accuracy* or *weighted relative accuracy*, it uses a heuristic that is based on the dispersion of the data. This algorithm also follows a different route by joining clustering approaches with predictive learning.

The R^2 system [Torgo, 1995] works to some extent analogously to other separate-and-conquer algorithms by selecting an uncovered region of the input data. But this selection differs from the mechanism used in regular separate-and-conquer learning. However, it also allows for rules to overlap and the rules predict linear models instead of a single target value.

Other mechanisms for learning regression rules are mainly based on ensemble techniques as used in the RULEFIT learning algorithm [Friedman and Popescu, 2008] or in REGENDER [Dembczyński *et al.*, 2008]. The first algorithm performs a gradient descent optimization, allows the rules to overlap, and the final prediction is calculated by the sum of all predicted values of the covering rules instead of that of a single rule. The second one uses a forward stagewise additive modeling.

[1] The famous RIPPER algorithm [Cohen, 1995], one of the most accurate rule learners for classification is also based on the separate-and-conquer paradigm.

Another popular technique to deal with a continuous target attribute is to discretize the numeric values as a preprocessing step and afterwards employ regular machine learning methods for classification. Research following this path can be found in [Torgo and Gama, 1996; Weiss and Indurkhya, 1995]. The main problem here is that the number of bags for the discretization process is not known in advance. For this reason the performance of this technique strongly depends on the choice of the number of classes.

3 Separate-and-conquer rule learning and Regression

Most inductive rule learning algorithms for classification employ a separate-and-conquer strategy for learning rules that allow to map the examples to their respective classes. The basic idea of the separate-and-conquer strategy [Fürnkranz, 1999] is to cover a part of the example space that is not explained by any rule yet (the conquer step). This region is covered by searching for a rule that fulfills some properties, i.e., has a low error on this partition of the input space. After this rule is found, it is added to a set of rules, and all examples that are covered by the rule are removed from the data set (the separate step). Then, the next rule is searched on the remaining examples. This procedure lasts as long as (positive) examples are left. The two constraints that all examples have to be covered (also called *completeness*) and that no negative example has to be covered in the binary case (*consistency*) can be relaxed so that examples remain uncovered in the data or negative examples are covered by the set of rules. This relaxation mostly is driven from preventing overfitting.

In the end, the algorithm returns a set of subsequently learned rules. For classification of unseen examples, each of the rules in the list is tested whether or not it covers the example. The first rule that covers the example (i.e., matches all the given attribute values) "fires" and predicts the value of the example by using the head of the rule. If no rule in the (decision) list covers the example, the prediction is given by a default rule that usually predicts the majority class in the data.

In the following we will have a closer look at the main step of the algorithm[2], namely how to navigate through the search space. Most of the algorithms build all possible candidate rules from the data by using all values for a given attribute and include these attribute-value pairs in a candidate rule. Thus, an attribute-value pair (a condition) is added to a given candidate rule which results in a refined candidate rule, i.e., a refinement of the former candidate rule. For nominal attributes these values are given from the data itself but for numerical attributes usually all possible splitpoints are used. The splitpoints are calculated as the mean between two adjacent (previously sorted) values.

Finally, when all candidates with one condition are generated a heuristic is used to determine the best one. Then, the best candidate rule is stored and refined to yield all refinements with two conditions. For nominal attributes the used ones are stored (and not used any more) and for numerical ones the relations $<$ and $\geq$ are evaluated. This means that a numerical attribute may occur twice in one rule by using it for a test on $<$ and on $\geq$. This procedure usually runs as long as negative examples are covered. In this step the algorithm also ensures that a minimum number of examples is covered (a user-given value). For all experiments (cf. Section 5 and 6) we fixed the minimum coverage to 3 examples.

Note that missing attribute values can never be covered and the attribute with the missing value is ignored. When the class of an instance is missing it is removed from the dataset in a preprocessing step. As search strategy simple hill-climbing was used.

There are many different heuristics to navigate the search (for an overview see [Fürnkranz and Flach, 2005]) but all of them are trying to maximize the coverage of positive examples (p) and to minimize the negative coverage (n). To reach this objective different ways are employed but usually, in some way, there has to be a combination of consistency (i.e., the error or the negative coverage) of the rule and its generality (i.e., the number of examples that are covered). Most of the heuristics have a fixed trade-off but some of them feature a parameter to adjust it. In previous work the parameters of some of these heuristics were tuned, so that they achieved the most accurate trade-off between consistency and coverage [Janssen and Fürnkranz, 2010a]. In this work we follow the same path by defining such a parametrized heuristic and by tuning its parameter to yield the best fit between these two objectives.

3.1 Separate-and-conquer for regression

As noted above some of the properties that come with categorical binary data do not apply for numerical target variables. Thus the algorithm had to be adapted in several ways. First of all, each evaluation of a single splitpoint requires a scan through the data. For this reason a novel splitpoint method has to be developed that allows using only a subset of all splitpoints to prevent the algorithm from getting too inefficient (cf. Section 3.3). Mechanisms for the efficient computation of splitpoints known from classification proved to be inefficient in our first experiments.

The heuristics that were introduced for the task of classification were not suitable for regression either. In Regression there is no notion for positive or negative examples. Hence, an alternative error measure has to be defined. The default rule also has to be adapted because there is no majority class any more. A simple way to do this is to take the mean over all remaining examples as prediction. Another way would be to take the mean of all examples. We experimented with both settings (cf. Section 6). Finally, the methods for evaluating the final model have to be adapted because using measures like *accuracy* (the percentage of correctly predicted examples) is not practicable any more.

3.2 Error measures for regression

There are several ways to compute the error of a rule or of a complete model for regression tasks. This section gives an overview of the measures we used.

In the following m denotes the total number of examples in the (current) dataset, y is the value of the current example, $\bar{y}$ is the value predicted by the rule, and y' is the mean over all examples.

The *mean absolute error* is the mean of the sum of the absolute errors of all examples that are covered by the rule

$$L_{MAE} = \frac{1}{m}\sum_{i=1}^{m}|y_i - \bar{y}_i| \tag{1}$$

The *root mean squared error* is defined by taking the root of the *mean squared error*

[2] For pseudo-code of the algorithm see [Janssen and Fürnkranz, 2010b].

$$L_{RMSE} = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(y_i - \bar{y}_i)^2} \quad (2)$$

The problem of L_{RMSE} is that it is domain-dependent. As the amplitude of the values in the domain is changing the amplitude in the error measures is changing as well. Thus, the errors are not comparable among different datasets. For using this measures to compute the heuristic value this may not be a problem because only candidate rules are compared to each other. But if a combination of the error and the coverage is taken this becomes crucial due to normalization issues.

For the normalization of the L_{RMSE} usually the deviation from the mean is used which is given by

$$L_{default} = \sum_{i=1}^{m}(y_i - y')^2. \quad (3)$$

Thus, the *relative root mean squared error*[3] becomes

$$L_{RRMSE} = \frac{L_{RMSE}}{\sqrt{\frac{1}{m} \cdot L_{default}}}. \quad (4)$$

These measures can be used for evaluating a single candidate rule but also for evaluating a whole theory (an ordered set of rules). Note that the L_{RRMSE} has its best value at 0 when each example is classified with the correct value. Theoretically, its worst value is 1 because only the value calculated by $L_{default}$ was used for predicting the value of an unseen example. Due to the split in 10 folds that happens during the cross-validation the values of the L_{RRMSE} can become bigger than 1. A stratification of regression values is not possible and this may results in splits where the test fold contains examples that share values that have never appeared in the training fold. In these cases the prediction with the value of $L_{default}$ would have been superior to the values predicted by the learned model. Note that some databases are practically even encode randomness because the L_{RRMSE} values on those where always bigger than 1 independently from the used algorithm.

To derive the *relative coverage* the number of covered examples divided by the total number of examples is taken.

$$relCov = \frac{1}{m} \cdot coverage(Rule). \quad (5)$$

We decided to combine the error and the generality of a rule by using the *rrmse* and the *relative coverage*

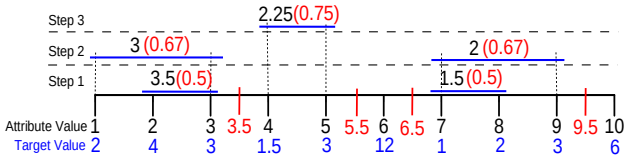
$$h_{cm} = \alpha \cdot (1 - L_{RRMSE}) + (1 - \alpha) \cdot relCov. \quad (6)$$

Here, the parameter α enables a trade-off between the error and the generality of the rule. For $\alpha = 1$ the relative coverage is ignored and thus the rules are evaluated solely by inspecting their error. This setting would yield a model that consists only of rules that cover a single example in the data and would thus clearly lead to overfitting[4]. The other extreme is to set $\alpha = 0$ which results in ignoring the error of the rule. A model built with this setting would only consist of the default rule, because its coverage is the highest that could be achieved by any rule. The optimal trade-off lies somewhere in between these two extremes.

[3] In the remainder of the paper abbreviated with *rrmse*.

[4] Note that this holds only in a scenario where a rule may cover a single example.

Figure 1: Example of the splitpoint clustering method



The heuristic h_{cm} is an adaptation of a previously introduced heuristic called *relative cost measure* [Fürnkranz and Flach, 2005]. Its formula is given by

$$h_{cr} = c_r \cdot \frac{p}{P} - (1 - c_r) \cdot \frac{n}{N}. \quad (7)$$

where p is the positive coverage of the rule, n is the negative coverage of the rule, P is the total number of positives and N is the total number of negatives.

It was designed for evaluating classification rules thus relying on coverage statistics. In previous work [Janssen and Fürnkranz, 2010a] an optimal setting for the parameter of h_{cr} was found ($c_r = 0.342$). It encodes a clear favor of the consistency (explained by $\frac{p}{P}$) over the coverage (denoted by $\frac{n}{N}$). It achieved good performance among different classification heuristics as shown in [Janssen and Fürnkranz, 2010a] since it was the second best heuristic of all. Thus the motivation to modify exactly this heuristic was the good performance and that it is best suited to be adapted to regression.

3.3 Splitpoint processing

As noted above, the generation of all possible splitpoints would be too costly. To avoid this a method to restrict the splitpoints for an attribute was developed. The basic idea comes from supervised clustering. Thus, we try to identify regions in the data of the current attribute that share a small error computed on the target variable. The aim of the clustering is to yield partitions of the attribute that share a low error in the hope that the error of a rule that covers these regions will also be low. Clustering stems from the same motivation because it also guarantees that each cluster has the lowest possible error. The user has to define how many clusters and hence how many splitpoints are desired. We experimented with different settings but surprisingly a rather low number of splitpoints seemed to be sufficient (cf. Section 5.1).

Figure 1 displays how the cluster algorithm works. In the example in Figure 1 the attribute has 10 values moving equidistantly from 1 to 10. The values depicted in blue are those of the target attribute of the respective example. In the first step the attribute values are ordered ascending and each value becomes a cluster containing exactly this value. Then two adjacent clusters are searched for which the error when using the mean of the two target values as prediction is the lowest. In the example these are the clusters 2 and 3, 7 and 8, and 8 and 9. Though the objective in the first step is to join two adjacent clusters both 2, 3 and 7, 8 are joined (its arbitrary whether to join 7 and 8 or 8 and 9). The mean of the first cluster is $3.5 = \frac{4+3}{2}$ and the second one has a mean of 1.5 (depicted in black in Figure 1 above the number ray). If the mean absolute error is taken, both clusters have an error of 0.5, which is shown in brackets and in red in the corresponding figure. An error of $0.5 = \frac{|4-3.5|+|3-3.5|}{2}$ is also the lowest error that can be achieved given the example data.

In the second step the function is executed recursively and again those clusters are joined that have the lowest error among all possible clusters. So, in this step, cluster 1 is joined with the second cluster and the cluster with a value of 9 is joined with the third cluster. The error of both clusters grows to 0.67 because adding the respective example does yield a raise of the error (i.e., $L_{MAE} = \frac{|2-3|+|4-3|+|3-3|}{3} = 0.67$ for the first cluster). Joining any of the untouched clusters leads to a higher error which means that the cluster with next lowest error is built in step 3. After the second step two clusters containing at least 2 examples were built and therefore 5 splitpoints exist. In the example the user given number of splitpoints is set to 4. Hence another cluster has to be built until the algorithm is finished. This last cluster is derived by joining the clusters with the values 4 and 5 and it yields an error of 0.75.

After the third step 3 clusters are built and the splitpoints are simply derived by taking the mean between the values of two adjacent clusters or two values if the cluster contains only one example. The 4 splitpoints are 3.5, 5.5, 6.5, and 9.5 (depicted in red in Figure 1 in the number ray). We have evaluated the effectiveness of the splitpoint method by comparing it to the usage of another splitpoint method where *n* splitpoints are selected equidistantly. The results of this comparison are shown in Section 6.1. For the computation of the error the *mean absolute error* was used. This choice is arbitrary but experiments with the *root mean squared error* did not yield any performance difference.

3.4 Parameters of the algorithm

There are 3 parameters the user has to specify.

- The parameter of the heuristic,
- the number of splitpoints (*splitpoint*-parameter), and
- the percentage of examples that are left uncovered (*left-out*-parameter).

The parameter of the heuristic is optimized with a greedy procedure that narrows down the region of interest. This procedure is described in detail in Section 5.

The number of splitpoints is crucial for the runtime of the algorithm. This value was optimized by testing different values (cf. Section 5.1).

The last user-given parameter is the percentage of examples that are left uncovered by the outer loop of the algorithm. This parameter clearly depends on the dataset. During the experiments there was some evidence that we had included databases that basically encode randomness and for those learning anything results in worse performance (e.g. the dataset *quake*).

4 Experimental setup

To optimize the 3 parameters some datasets were used for tuning and were split into 2 folds of equal size. On the first fold of each dataset, all steps of the optimization procedure were done and afterwards the best model was evaluated on the second fold. This is also done vice versa. Hence, the experiments yield two configurations of the same algorithm that only differ in the parametrizations. A test of the parametrizations on the hold-out folds of the tuning datasets is the first step of the evaluation. Additionally some insights are gained by evaluating the two variants also on those datasets that were used during the optimization. To complete the evaluation, the two resulting configurations were also evaluated on some datasets that were not used for any optimization purposes.

The aim of the experiments was to optimize the parameters of the algorithm on a set of diverse datasets to capture characteristics of a wide variety of different datasets. Our hope was that by taking a set of datasets that are very different the parameters would be more stable. For this reason, we selected 29 databases in total from the UCI-Repository [Asuncion and Newman, 2007] and from Luis Torgos website[5]. The datasets were divided into 20 sets that were used during the tuning phase and 9 sets that were only used for evaluation purposes. The tuning datasets were

> *abalone*, *auto-mpg*, *auto-price*, *breast-tumor*, *compressive*, *concrete-slump*, *cpu*, *delta-ailerons*, *echo-month*, *forest-fires*, *housing*, *machine*, *pbc*, *pyrim*, *quake*, *sensory*, *servo*, *strike*, *triazines*, *winequality-white*

As mentioned above the main motivation to select these datasets was to capture a lot of different learning problems. Thus, the number of nominal and numerical attributes should be different among the databases and the domains from which they origin should be as diverse as possible.

The 9 datasets that were used to evaluate the algorithms were

> *auto93*, *auto-horse*, *cloud*, *delta-elevators*, *meta*, *r_wpby*, *stock*, *veteran*, *winequality-red*

The distribution among the 20 tuning databases in terms of nominal and numerical attributes as well as in terms of size should be approximately the same as in the testing datasets. Therefore both bags of data contain some small, some medium and some big databases. For a detailed overview of the datasets see [Janssen and Fürnkranz, 2010b].

4.1 Evaluation methods

The primary method to evaluate the algorithm was the *rrmse*. The advantage of this evaluation measure clearly lies in its domain-independency. For some of the experiments it would take too much space to include results on every single dataset. In those cases our means for evaluating the different algorithms was to average the results over all datasets. We are aware of the problems that come with averaging results over many different domains (i.e., some databases may be outliers with huge variance compared to the majority of the other datasets) and hence include a Friedman-Test with a post-hoc Nemenyi-Test as suggested in [Demsar, 2006]. The resulting CD-charts give insights how good the algorithms perform by evaluating their ranking independently from using average accuracy.

There are other ways to evaluate regression algorithms domain-independently. The correlation coefficient for instance is also widely used. But there are some drawbacks from using this method regarding rule learning algorithms. For results including the correlation coefficient and a discussion of the drawbacks see [Janssen and Fürnkranz, 2010b].

5 Optimizing parameters

For the split into the 2 folds, all datasets were randomized in advance using the unsupervised randomize function of *weka* [Witten and Frank, 2005]. All evaluation measures were computed using one run of a 10-fold cross validation.

[5]These databases can be downloaded at `http://www.liaad.up.pt/~ltorgo/Regression/DataSets.html`.

Table 1: Results for the splitpoint computation and left-out-parameter (average *rrmse* over the 5 parametrizations of the heuristic)

parameter (splitpoint)	folds 1	folds 2	parameter (left-out)	folds 1	folds 2
1	1.0675	1.0540	0	0.9929	1.0209
3	**0.9929**	1.0256	0.01	0.9787	1.0221
5	1.0132	1.0261	0.02	0.9776	1.0182
7	1.0067	1.0245	0.03	0.9759	1.0156
9	0.9992	**1.0209**	0.05	0.9739	0.9940
11	1.0126	1.0427	0.1	**0.9704**	0.9835
19	1.0163	1.0240	0.2	0.9736	**0.9701**

5.1 Optimization of the splitpoint and the left-out parameter

Though these two parameters are likely to have a small deviation in performance among different databases, we decided to optimize them first and fix them before we start optimizing the parameter of the heuristic. We believe that the parameter of the heuristic has a stronger influence on the performance of the algorithm than the other two parameters. This is mostly because the heuristic is used to evaluate every single candidate rule and therefore is the most important factor in the algorithm. Note that the heuristic also has a strong influence on the quality of the rules and on the total number of rules found by the algorithm. The other two parameters are also influencing the performance of the algorithm but rather in an indirect way by assuring to provide splitpoints of good quality and by leaving those examples untouched that are hard to learn.

For this reason, we focussed more on the heuristic parameter than on the other two. If we had concentrated more on these two the performance of the algorithm could have been become a bit better but our main idea was to derive stable parameters for the heuristic and we believe that the gain in performance depends stronger on the heuristic parameter than on the other two (cf. the experimental results in section 5.2).

To start optimizing the splitpoint parameter the other two had to be fixed. In advance it is not known how to determine these values. Thus, the *left-out*-parameter was fixed to 0, therefore all examples have to be covered. In this case the default rule is built by using the mean of all examples. In other cases where examples remain uncovered it is built by using the mean of all uncovered examples. On the contrary, it is not obvious what parameter value can be used for the heuristic. For this reason, 5 different values were used during the optimization. To make a choice, the two extremes were included ($\alpha = 0$, and $\alpha = 1$), and some values in between, namely 0.4, 0.5, and 0.6. These values were used to include different preferences of the heuristic. Clearly, using only two parameters would be suboptimal because there is some evidence that the optimal parameter rather would lie somewhere in the middle of the domain than at the beginning or the end of it [Janssen and Fürnkranz, 2010a]. We expected the curve yielded by plotting the parameter over the error to be shaped like a U, where the two extreme values would results in a rather bad performance and the optimal value lies somewhere around 0.5 (cf. Section 5.2). To have a combined error measurement for the optimization procedure the mean over the *rrmse* of these choices was taken.

In the beginning, the *left-out*-parameter was fixed to a value of 0 yielding a starting point for the optimization of the *splitpoint*-parameter. To find the best value some intuitive values (1, 3, 5, 7, 9, 11, and 19) were used. All values bigger than 19 were skipped because a clear gain in runtime performance should be achieved. Using huge values would result in practically using all possible splitpoints and thus would not improve the algorithm's runtime[6].

Table 1 (left table) shows the results for the two optimization procedures (for the two folds of the tuning datasets). As can be seen the best number of splitpoints was 3 on the first folds and 9 on the second folds (the lowest error is depicted in bold in the figure). On the first folds, however, using 9 splitpoints yields the second best *rrmse* which lacks only 0.0063 behind the best performing number of splitpoints. On the second folds, using 9 splitpoints performed best followed by using 19 splitpoints. Using 3 splitpoints lacks 0.0047 in terms of *rrmse* behind the best one and therefore is the fourth best method. Nevertheless, the gap between different parametrizations seems to be bigger on the first folds than on the second ones. Regarding the split of all tuning datasets into 2 folds, these results seem to reflect the randomness in splitting the datasets.

After the optimization of the splitpoint parameter the same procedure was employed to the *left-out*-parameter of the algorithm. Here, the splitpoints were already fixed to 3 for the algorithm tuned on the first folds and to 9 for the variant tuned on the second folds. To find the best value also some intuitive parameters were used. Thus, the values 0, 0.01, 0.02, 0.03, 0.05, 0.1, and 0.2 were tested during this optimization. The setting where all examples are covered by rules was included to make sure that is more effective to leave some parts of the data uncovered. Clearly, an optimal setting is dataset-dependent. But it also depends on the quality of the induced rules. For numerical target variables it can be useful to cover only those parts of the data that share some common characteristics. For the remainder of the data it could be beneficial to treat them independently from their characteristics.

As can be seen in Table 1 (right table) two different parameters performed best on the two folds. Practically this can be attributed to the same reasons that were already discussed during the optimization of the splitpoint parameter. Thus, on the one hand the randomized split of the data into 2 folds of equal size could have manipulated the characteristics of the datasets. On the other hand it could also be possible that there is no unique best value for leaving examples uncovered. The results also show that leaving examples uncovered is mandatory for the performance of the algorithm.

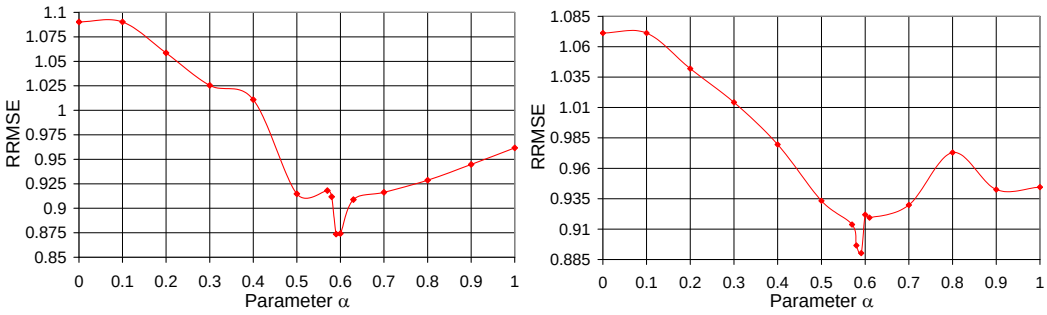
5.2 Optimization of the heuristics parameter

For the optimization of the heuristics parameter a framework similar to the one introduced in [Janssen and Fürnkranz, 2010a] was used. It employs a binary search to find the best parameter and was proven to yield stable parameters for classification heuristics as shown in [Janssen and Fürnkranz, 2010a].

The search is started with a range of intuitively appealing parameters. Thus, the two extremes of 0 and 1 are tested together with some values in between $(0.1, 0.2, ..., 0.9)$. All settings are evaluated by taking the average of the *rrmse* on the 20 datasets presented in Section 4. Then the best performing parameter is used for further inspection. Therefore, an area around this parameter is inspected in more

[6]Note that the number of disjunct values for an attribute in the data is rather small.

Figure 2: Parameters over *rrmse* for both folds of the tuning datasets



(a) tuned on folds 1 (b) tuned on folds 2

detail. There are several choices to do this, but we decided to evaluate 6 parameters around the best one. Those are distributed equidistantly around the best parameter with decreasing the step size from 0.1 to 0.01. This procedure is executed recursively, so in the next step the 6 parameters around the next best value are tested. The search stops if the *rrmse* improvement falls below a threshold of $t = 0.0005$. This choice was arbitrary but we believe that the effort that has to be made to narrow down the parameter for the next step of the search procedure is too high compared to the performance gain the next execution may yield.

Figure 2 shows a graphical interpretation of the search for both experiments. For both of them very low parameter settings result in bad performance. When the parameters are increased the performance becomes better as long as the optimal setting is reached. After that it decreases again.

For the parameters that are optimized on the first folds of the datasets (Figure 2 (a)) the curve shows some fluctuations in the part located left of the best parameter. In spite of this behavior the curve depicted in (Figure 2 (b)) is monotonically decreasing in this area. For parameter settings that are bigger than the best parameter the curve in the left figure is now showing a monotone increase whereas it shows more fluctuations when the parameter is tuned on the second folds of the partitioned datasets.

Interestingly, the best parameters are very similar in both experiments. This means that the parameters are stable among different splits of the datasets. On the first folds of these datasets the best parameter lies at 0.59 and on the second folds it was 0.591. For the first folds the parameter 0.591 lacks only 0.007 behind in terms of *rrmse*. For the second folds the difference in performance was 0.001.

Assumed that the best parameter lies somewhere in the region of 0.6, consistency should be preferred over coverage for regression rules. This also holds for classification rules where the preference of consistency is even stronger than in regression. Nevertheless it is an interesting result that the evaluation of a rule's quality follows similar standards in classification and in regression.

6 Results

6.1 Splitpoint processing

Table 2 shows a comparison of the runtime of 2 different splitpoint methods. At first, 3 equidistant splitpoints per attribute were used. Then, 3 clustered splitpoints were employed. Evaluating all splitpoints was too costly[7]. All runtimes depicted in Table 2 are the averages of 10 independent runs on a dual Pentium 4 2.8 GHz processor with 2 GB RAM on the 9 datasets used for testing (cf. Section 4).

Table 2: Runtime of different splitpoint methods on the test set

method	runtime (in sec.)
3 equidistant splitpoints	2625.4
3 clustered splitpoints	1234.3

As can be seen in Table 2 the clustered splitpoint computation is more efficient than the equidistant method. At first sight this may appear contrary to what could be expected. Due to the much more simpler computation of equidistant splitpoints this method should be faster than the clustering method. But note that this evaluation was done by letting the whole algorithm run on the 9 test datasets. Not surprisingly the quality of the equidistant splitpoints is worse compared to the clustered splitpoints. This results in a significantly higher number of candidate rules that have to be evaluated during the search for the best rule which can be drastically reduced by using clustered splitpoints.

6.2 Comparison with other systems on the tuning datasets

The main focus of the comparison is how well the algorithm performs against other regression algorithms. Table 3 gives an overview of the different algorithms compared to each other on the two folds using the *rrmse*. From now on the tuned algorithm is referred by the name *SeCoReg*.

There are 4 other algorithms that are all implemented in *weka* [Witten and Frank, 2005] which were used to compare our system with. Clearly, some of them are much more complex than our rather simple algorithm[8]. On the other hand most of them employ more complex models, i.e., hyperplanes like the *multilayer perceptron* (MLP) or support vectors like the *SVMReg*. The *linear regression* (Linear Reg.) is also a rather simple algorithm that nevertheless employs quite a good trade-off between runtime and error. *M5Rules* uses rules to explain the data. These rules predict linear models which makes the algorithm much more flexible because each rule is able to map the examples on many different outcome values.

The parameters of all *weka* algorithms were left at default values. The reasons to select these 4 algorithms were that our implementation had to prove that it is comparable

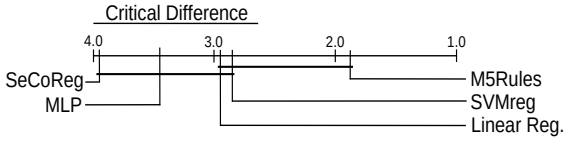
[7] This is due to some huge datasets.

[8] Note that the algorithm neither has a pruning functionality nor an optimization phase.

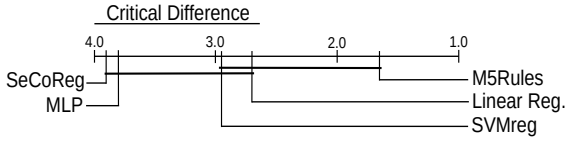
Table 3: Results in terms of average *rrmse* for different algorithms on the tuning datasets

	folds 1	folds 2
M5Rules	0.7425	0.8058
Linear Reg.	0.8145	0.9116
MLP	1.0154	1.3890
SVMreg	0.7917	0.8500
SeCoReg (folds 1)	*0.8736*	0.9291
SeCoReg (folds 2)	0.8976	*0.8903*

Figure 3: Comparison of all algorithms against each other with the Nemenyi test. Groups of algorithms that are not significantly different (at $p = 0.05$) are connected.



(a) folds 1



(b) folds 2

in terms of error to other state-of-the-art systems. Another reason to select these particular algorithms for benchmark was the lack of freely available regression rule learning algorithms. The only free system we found was REGENDER and a comparison is given in Section 6.3.

In Table 3 the results of all algorithms on the two folds of the tuning datasets are displayed. Results of both derived *SeCoReg*-algorithms are shown together with their performance on the data sets on which they were tuned (in italics). Not surprisingly both variants of the algorithm that were tuned on the respective folds are better than using them on the left-out folds. The ranking of the algorithms is similar on both experimental variants. The best one was *M5Rules* followed by the *SVMreg* and the *linear regression*. The *SeCoReg* was ranked on the 4th place in both experiments (by average *rrmse*), only slightly behind the *linear regression* (lacking 0.0831 behind on the first folds and 0.0175 on the second folds). The *Multilayer Perceptron* had the worst performance with a rather big gap to the next better algorithm.

Figure 3 shows CD-charts for both experiments. Note that the figure displays ranks averaged on all datasets. Only the algorithm *M5Rules* was significantly better than the *SeCoReg* in both cases.

6.3 Comparison with other algorithms on the test sets

To validate the results on completely different datasets the algorithm was also tested on 9 independent test sets (cf. Section 4). This step is necessary to make sure that the tuning datasets, even though they were split into two disjunct folds, were not overfitted during the parameter tuning phase. Table 4 displays the results in terms of *rrmse* on the test databases for all of the 4 weka algorithms and the two configurations of the *SeCoReg*-learner. The ranking of the algorithms differs slightly compared to the results on the 20 datasets. Hence, on the test sets the *SVMreg* performs best followed by the *M5Rules*-system. On the third place the first *SeCoReg*-learner appears. It was only slightly worse in performance compared to the *M5Rules*-learner. The next best algorithm is the second *SeCoReg*-learner which has achieved a marginal better *rrmse* than the *linear regression*. As in the previous experiments the *multilayer perceptron* was the worst algorithm.

Table 5: Results in terms of *rrmse* compared to *RegENDER* on 7 datasets of the test set

algorithm	avg. rrmse	avg. rank
SeCoReg (folds 1)	0.8154	3.00
SeCoReg (folds 2)	0.8538	3.13
RegENDER (10 rules)	0.9008	3.88
RegENDER (100 rules)	0.9291	3.88
RegENDER (# rules from folds 1)	0.9221	3.75
RegENDER (# rules from folds 2)	0.9034	3.38

Thus, on the test sets the tuned *SeCoReg*-algorithm achieved better results than in the previous experiment. Here, the best configuration of the algorithm is ranked in third place. Note that the dataset *meta* shows huge standard deviations for some algorithms (*M5Rules*, *linear regression* and *MLP*). We attribute this to the separation of the data into the 10 folds of the cross validation.

Additionally to the error measurements a Friedman-Test was employed like in the previous Section (cf. Section 6.2). Contrary to the prior results, the Friedman-Test was not rejected at a *p*-value of 0.05 (the critical *F*-value was 2.196 but to reject the test it had to be bigger than 2.492). It would have been rejected at a *p*-level of 0.1, but this was not significant enough to include these results in the paper. For this reason the Nemenyi-Test could also not be done on the test sets. Practically, this means that the *SeCoReg* algorithm does not differ significantly from the 4 weka algorithms at a significance level of 0.05.

Table 5 shows a comparison to *RegENDER* [Dembczyński *et al.*, 2008]. The dataset *auto-horse* contains missing class values which cannot be handled by *RegENDER*. Therefore, this dataset was left out. In addition the results on the dataset *meta* showed strong fluctuations as mentioned before. For this reason this dataset was also left out. *RegENDER* has a parameter to specify the number of rules in the ensemble. To make a choice the algorithm was tested with 10 and 100 rules and with the same number of rules the two *SeCoReg* variants had found on each test set. Clearly, using more rules will result in a lower error (cf. [Dembczyński *et al.*, 2008]) but we think it is fair to run the algorithm with the same number of rules as used in the *SeCoReg*-learner.

The *SeCoReg*-algorithm was slightly better in average *rrmse* and the average rank was also better. Nevertheless, a Friedman Test was rejected ($p = 0.05$) but the Nemenyi Test showed that all algorithms were in the same equivalence class (the critical distance extends over all algorithms) and therefore do not differ statistically significant.

To sum up, both tuned variants of the presented algorithm are not able to beat state-of-the-art systems. They are rather situated in the middle of the performance of the other algorithms (this holds at least for the test sets). Especially on the 9 test sets it became clear that the *SeCoReg* rule learners are able to achieve a performance comparable to the results of the 4 *weka* algorithms and *RegENDER*.

Table 4: Results in terms of *rrmse* for different weka algorithms and the *SeCoReg*-learners on the test set

dataset	SVMreg	M5Rules	Linear Reg.	MLP	SeCoReg (tuned on folds 1)	SeCoReg (tuned on folds 2)
auto-horse	0.32± 0.08	0.37± 0.14	0.32± 0.11	0.34± 0.10	0.52± 0.18	0.61± 0.11
auto93	0.66± 0.12	0.58± 0.19	0.67± 0.20	0.57± 0.19	0.65± 0.17	0.85± 0.29
cloud	0.39± 0.12	0.42± 0.16	0.40± 0.13	0.62± 0.33	0.61± 0.19	0.67± 0.15
delta-elevators	0.61± 0.01	0.60± 0.01	0.61± 0.01	0.63± 0.01	0.78± 0.03	0.77± 0.03
meta	0.92± 0.08	1.86± 1.58	2.33± 1.72	1.40± 0.90	1.00± 0.02	1.01± 0.03
r_wpbc	1.03± 0.16	1.14± 0.19	1.04± 0.13	2.20± 0.56	1.35± 0.20	1.27± 0.18
stock	0.37± 0.05	0.14± 0.03	0.36± 0.04	0.20± 0.04	0.25± 0.03	0.26± 0.04
veteran	0.93± 0.15	1.23± 0.61	1.07± 0.36	3.01± 1.78	1.09± 0.22	1.21± 0.33
winequality-red	0.82± 0.03	0.81± 0.03	0.81± 0.03	0.95± 0.08	0.98± 0.09	0.95± 0.04
averages	0.6739	0.7942	0.8456	1.1017	0.8040	0.8438

Due to the rather simple design of the *SeCoReg*-algorithm these results seem to be promising.

7 Conclusion and further work

In this paper a new rule learning algorithm for the task of regression was presented. It was shown that the algorithm performs comparable to different state-of-the-art algorithms implemented in *weka* and *RegENDER*, a rather new algorithm.

A new splitpoint generation method was introduced. This method proved to support the quality of candidate rules and even results in lower runtime compared to naive methods like the generation of equidistant splitpoints. Nevertheless, the number of generated candidate rules directly depends on the number of splitpoints. But as shown in the experiments at least for one configuration of the algorithm a number of 3 splitpoints per numerical attribute was enough.

A novel rule learning heuristic was introduced that clearly improves the algorithms performance due to its flexibility in weighting the error of a rule with its coverage. An optimal setting for this regression rule heuristic was presented and it proved to be stable since the parameter values are nearly the same. An interesting observation is that, as known from classification, in regression the rules consistency also should be preferred over its coverage.

A promising path to optimize the algorithm would be to adapt the advantages of algorithms like *M5Rules* which predicts linear models in the head of each rule. On the one hand the performance of the algorithm should be drastically improved when using linear models instead of single target values. On the other hand, much of the interpretability of the rule set would be lost when doing so.

Acknowledgments

This research was supported by the *German Science Foundation (DFG)* under grant no. FU 580/2-2.

References

[Asuncion and Newman, 2007] A. Asuncion and D.J. Newman. UCI machine learning repository, 2007.

[Clark and Niblett, 1989] Peter Clark and Tim Niblett. The CN2 induction algorithm. *Machine Learning*, 3(4):261–283, 1989.

[Cohen, 1995] William W. Cohen. Fast effective rule induction. In *ICML*, pages 115–123, 1995.

[Dembczyński *et al.*, 2008] Krzysztof Dembczyński, Wojciech Kotłowski, and Roman Słowiński. Solving regression by learning an ensemble of decision rules. In *ICAISC '08*, pages 533–544, Berlin, Heidelberg, 2008. Springer-Verlag.

[Demsar, 2006] Janez Demsar. Statistical comparisons of classifiers over multiple datasets. *Machine Learning Research*, 7:1–30, 2006.

[Demšar, 1999] D. Demšar. Obravnavanje numericnih problemov z induktivnim logicnim programiranjem. Master's thesis, Faculty of Computer and Information Science, University of Ljubljana, Slovenia, 1999. In Slovene.

[Friedman and Popescu, 2008] Jerome H. Friedman and Bogdan E. Popescu. Predictive learning via rule ensembles. *Annals Of Applied Statistics*, 2:916, 2008.

[Fürnkranz and Flach, 2005] Johannes Fürnkranz and Peter Flach. ROC 'n' rule learning – Towards a better understanding of covering algorithms. *Machine Learning*, 58(1):39–77, 2005.

[Fürnkranz, 1999] Johannes Fürnkranz. Separate-and-conquer rule learning. *Artificial Intelligence Review*, 13(1):3–54, February 1999.

[Holmes *et al.*, 1999] Geoffrey Holmes, Mark Hall, and Eibe Frank. Generating rule sets from model trees. In *Twelfth Australian Joint Conference on Artificial Intelligence*, pages 1–12. Springer, 1999.

[Janssen and Fürnkranz, 2010a] Frederik Janssen and Johannes Fürnkranz. On the quest for optimal rule learning heuristics. *Machine Learning*, 78(3):343–379, March 2010. DOI 10.1007/s10994-009-5162-2.

[Janssen and Fürnkranz, 2010b] Frederik Janssen and Johannes Fürnkranz. Separate-and-conquer regression. Technical Report TUD-KE-2010-01, TU Darmstadt, Knowledge Engineering Group, 2010.

[Karalič and Bratko, 1997] Aram Karalič and Ivan Bratko. First order regression. *Machine Learning*, 26(2-3):147–176, 1997.

[Quinlan, 1992] Ross J. Quinlan. Learning with continuous classes. In *5th Australian Joint Conference on Artificial Intelligence*, pages 343–348, Singapore, 1992. World Scientific.

[Torgo and Gama, 1996] Lus Torgo and Joo Gama. Regression by classification. In *In Proceedings of SBIA96, Borges*, pages 51–60. Springer-Verlag, 1996.

[Torgo, 1995] Luis Torgo. Data fitting with rule-based regression. In *In Proceedings of the 2nd international workshop on Artificial Intelligence Techniques (AIT'95*, 1995.

[Ženko *et al.*, 2005] Bernard Ženko, Saso Džeroski, and Jan Struyf. Learning predictive clustering rules. In *In 4th Intl Workshop on Knowledge Discovery in Inductive Databases: Revised Selected and Invited Papers, volume 3933 of LNCS*, pages 234–250. Springer, 2005.

[Wang and Witten, 1997] Y. Wang and I. H. Witten. Induction of model trees for predicting continuous classes. In *Poster papers of the 9th European Conference on Machine Learning*. Springer, 1997.

[Weiss and Indurkhya, 1995] Sholom M. Weiss and Nitin Indurkhya. Rule-based machine learning methods for functional prediction. *Journal of Artificial Intelligence Research*, 3:383–403, 1995.

[Witten and Frank, 2005] Ian H. Witten and Eibe Frank. *Data Mining — Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann Publishers, 2nd edition, 2005.

Kernelized Map Matching for Noisy Trajectories

Ahmed Jawad , Kristian Kersting
Knowledge Discovery Dept., Fraunhofer IAIS, 53754 Sankt Augustin, Germany
ahmed.jawad@iais.fraunhofer.de , kristian.kersting@iais.fraunhofer.de

Abstract

Map matching is a fundamental operation in many applications such as traffic analysis and location-aware services, the killer apps for ubiquitous computing. In the past, several map matching approaches have been proposed. Roughly speaking they can be categorized into four groups: geometric, topological, probabilistic, and other advanced techniques. Surprisingly, kernel methods have not received attention yet although they are very popular in the machine learning community due to their solid mathematical foundation, tendency toward easy geometric interpretation, and strong empirical performance in a wide variety of domains. In this paper, we show how to employ kernels for map matching. Specifically, ignoring map constraints, we first maximize the consistency between the similarity measures captured by the kernel matrices of the trajectory and relevant part of the street map. The resulting relaxed assignment is then "rounded" into an hard assignment fulfilling the map constraints. On synthetic and real-world trajectories, we show that kernels methods can be used for map matching and perform well compared to probabilistic methods such as HMMs.

1 Introduction

Map matching is a fundamental operation for many real-world applications that are currently changing how we as a society use computers in daily life. Following [Mitchell, 2009]: after decades of analysing "historical" data, 'location-based services' are a major driving force for analysing "real-time" and "real-space" data that record personal activities, conversations, and movements in an attempt to improve human health, guide traffic, and advance the scientific understanding of human behaviour. Therefore, it is not surprising that map matching — the problem of mapping a temporal sequence of coordinates onto a given network — has received a lot of attention.
Roughly speaking existing approaches for map matching can be categorized into four groups: geometric, topological,probabilistic, and other advanced techniques. Surprisingly, however, kernel methods such as support vector machine and Gaussian processes have not received attention yet although they are very popular in the machine learning community due to their solid mathematical foundation, tendency toward easy geometric interpretation, and strong empirical performance in a wide variety of domains. In a nutshell, kernel methods first process a dataset into a kernel matrix that roughly expresses the idea that two data points are "equivalent as far as some function f of the data can tell". By representing the data in terms of a kernel matrix, the data can be of various types, and also heterogeneous types such as trees and graphs. This makes kernel approaches very flexible. In a second step, a variety of kernel algorithms that have been developed can be used to analyse the data, using only the information contained in the kernel matrix. Kernels are, also, readily extendible therefore every kernel matrix provides an opportunity to integrate its knowledge with existing kernels in the field. An investigation of using kernel methods for map matching motivated by their well-known strength was the seed that grew into our proposal for kernelized map matching. Through this article, we make the following contributions in map matching research:

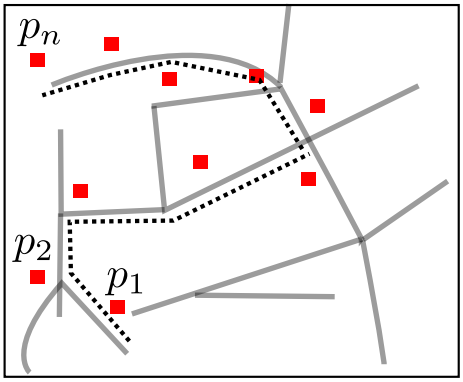
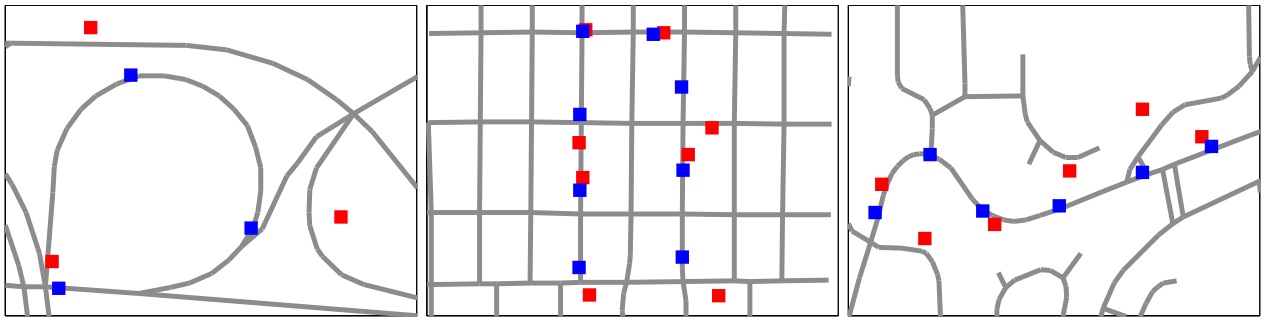


Figure 1: illustration of the map matching problem: Given a graph $G = (V, E)$, denoted as gray edges, and a trajectory $T = \{p_1, p_2, ..., p_n\}$, denoted as red squares, find the ground truth path (dashed line) starting from T.

1. Establishing a link between 'map matching' and 'kernel methods'.
2. Specifically building and investigating an instantiation of this link: a simple yet effective kernelized map matching approach, called 'Kernelized Map Matching' (KMM).
3. To do so, we also develop a simple kernel to capture spatio-temporal correlations under one umbrella.

Specifically, triggered by the observation that matching a trajectory of coordinates would be easy if the observed coordinates were noise-free — the coordinates would simply constitute the solution — one may propose to treat the map matching problem as a regression problem. That is, we treat a trajectory as a function t for which we observe a



(a) KMM: over a roundabout (b) KMM: over grid structure (c) KMM: Robustness to noise

Figure 2: illustrations of KMM's performance on challenging real-world situations. The red points show the input trajectory and the blue points the output path by KMM. In all three cases, KMM recovers the exact groundtruth paths. (coloured)

sequence of noisy values, the coordinates $t(i) + \epsilon$ at inputs $i = 1, 2, 3, \ldots, k$, the temporal order of coordinates. The task is now: estimate the noise-free function t from the noise observations. In contrast to most traditional regression tasks, however, outputs are structured due to the physical constraints in the world and in turn there are non-linear dependencies among coordinates. Although kernel methods are powerful tools for modelling non-linear dependencies, and hence seem to be relevant for map matching, most kernel (regression) models focus on the prediction of a single output. Although generalizations to multiple outputs can often be achieved by training independent models for each one or tying parameters across dimensions, this fails to account for output correlations [Weston and Vapnik, 2002]. Consequently, we propose a different approach, namely to "embed" the output of F, i.e., the coordinates of trajectory into the same space as the trajectory and the network and in turn reducing the noise while still capturing the multi-output, non-linear dependencies present in trajectories. Specifically, ignoring map constraints, we first embed the trajectory and hence reduce noise. The resulting relaxed assignment is then "rounded" into an hard assignment fulfilling the map constraints. On synthetic and real-world trajectories, we show that this approach, called kernelized map matching, can be used for map matching and performs well compared to state-of-the-art hidden Markov models. Fig. 2 shows KMM performance over some of the tasks considered difficult among map matching experts [Newson and Krumm, 2009].

We proceed as follows. We start off by reviewing related work and mathematical background. In Section 3, we then introduce kernelized map matching. Before concluding, we present the results of our experimental evaluation in Section 4.

2 Related work:

The work presented in this paper is conceptually built upon three areas of research, namely map matching, structural embeddings and kernel methods. We briefly describe the background work in each of these areas respectively.

Quddus *et al.* [Quddus *et al.*, 2007] provide a good survey of existing map matching algorithms. Following them, existing approaches can be broadly categorized into four categories (or a combinations of them): geometric [Fox *et al.*, 2003], topological , probabilistic [Quddus *et al.*, 2007] and other advanced techniques [Taylor *et al.*, 2001]. Specifically, geometric map matching approaches utilize the shape of the spatial road network without considering the continuity or connectivity of it. Topological approaches use the connectivity and other topological features to restrict the candidate matches for solution points. Finally, probabilistic and advanced approaches are referred to as those "using more refined concepts such as a Kalman Filter, a fuzzy logic model or the application of hidden Markov model". Another dimension along which map matching approaches can be distinguished is "incremental vs. global" [Lou *et al.*, 2009].

In the present paper, we take the traditional, data-driven view of map matching (refer to [Lou *et al.*, 2009] for more details) which is defined as

> **Given** *a trajectory T and a street network G,* **find** *a path P in G that matches T with its real or ground truth path.*

Next to traditional map matching approaches, our approach builds upon ideas from structural embeddings [Quadrianto *et al.*, 2009], [Guo *et al.*, 2008]. The idea is to view trajectories as a special type of graph called Euclidean graph [Pemmaraju and Skiena, 2003].

Definition 1 (Euclidean graph): *A Euclidean graph $G(V, E)$ is a graph in which*

1. *the vertices represent points in the Euclidean plane $\mathbb{R}^2$, and*
2. *the edges are assigned lengths equal to the Euclidean distance between those points, i.e., a straight line between corresponding vertices.*

In order to view a trajectory as an Euclidean graph, we need to capture the spatial and temporal aspects of the trajectory inside it. The spatial aspect of a trajectory is already captured in the Euclidean graph through $\mathbb{R}^2$ coordinates of the graph's vertices. In order to capture the temporal aspects of a trajectory, we time stamp the vertices of the graph in the order of the trajectory. The poly-line structure of street network graph with its vertices embedded in $\mathbb{R}^2$ is, indeed, an Euclidean graph. To make it explicit, we can add additional vertices to every corner of the lines constituting a road segment. Consequently, map matching, can be viewed as a problem of

> "*matching, i.e., finding similarity between two Euclidean graphs*".

Indeed, Euclidean graphs can actually be found in many machine learning problems such as map matching [Brakatsoulas *et al.*, 2005], shape analysis, protein structure analysis, time series similarity analysis, among others. However, —to the best of our knowledge— ,the problem of matching Euclidean graphs, i.e., "finding similarity between two Euclidean graphs" has not been considered yet. Instead,

the problem has been approximated by casting it into a traditional "embedding" respectively "graph matching" problem and in turn dropping important information. Specifically, *graph matching* is typically formulated as a problem where we only seek a node-to-node matching between the input and output graphs. Euclidean graph matching is different as the nodes of the input graph (trajectory) can be mapped to any point lying on the edges of the output graph (street network graph).Embedding [Lee and Verleysen, 2007], on the other hand, is a generic problem: find a set of points in a (typically) low-dimensional space having similar properties and relationships as the points in the original input space. So far, however, it has only been considered for matching graphs in the traditional sense discussed above, see e.g. [Lee and Verleysen, 2007], [Guo *et al.*, 2008; Quadrianto *et al.*, 2009].Hence, this approach suffers from the same issues as the standard graph matching approach. Nevertheless, they employ kernel methods. We use a simple embedding technique called Multidimensional scaling(MDS) [Cox and Cox, 2000] defined as following:

Definition 2 Multidimensional Scaling(MDS)*: MDS is an embedding technique which uses the so-called stress function (the sum of squared differences between the similarity criteria of input and latent spaces) to come up with embedding X for input Y. Lets say K_x and K_y denote the similarity matrices for output X and input Y. Then the objective function F_o using MDS can be described as follows:*

$$F_o = \underset{X}{\operatorname{argmin}} \sum_{i,j} \left(K_x(i,j) - K_y(i,j)\right)^2 \qquad (1)$$

Embedding approaches i.e, MDS and other, are called 'kernelized embedding' [Guo *et al.*, 2008] when they use kernel matrices as similarity criteria. Kernel Methods help with embedding in special cases because they are more suitable to non-vectorial data sets (text, images, protein sequences, graphs, trajectories, etc..) [Shawe-Taylor and Cristianini, 2004]. Our method is inspired by these techniques but generalizes them to the Euclidean graph case.

3 Kernelized Map Matching

Recall from the introduction that map matching would be easy if the observed coordinates were noise-free. In this case the observed coordinates would simply constitute the solution. In general, we cannot expect to reduce the noise completely. Consequently, we propose a two steps approach:
(1) To reduce the noise, embed the trajectory from $\mathbb{R}^2$ into $\mathbb{R}^2$ using kernel methods to capture the multi-output, nonlinear dependencies present in trajectories.
(2) To account for remaining noise, "round" the embedding into an hard matching. We will now explain each of the steps in turn.

3.1 Euclidean Graph Matching

We proceed with some notation and problem description. $G(V,E)$ denotes a trajectory (input graph) where vertices are indexed by time t , i.e. $V = \{v_t\}_{t=1}^{|V|}, v_t \in \mathbb{R}^2$, while $G^{'}(V^{'},E^{'})$ denotes street network graph (output graph) where $V^{'} = \{v_i^{'}\}_{i=1}^{|V^{'}|}, v_{i'} \in \mathbb{R}^2$ are the set of nodes for street segments and $E^{'} = \{e_{i,j}^{'}\}_{i,j=1}^{|V^{'}|}, e_{i,j}^{'} \in \{\theta v_i + (1-\theta)v_j, \theta \in [0,1]\}$, if $e_{i,j}^{'}$ corresponds to a street segment otherwise it is empty. Notice that $e_{i,j}^{'}$ is defined as a convex combination of vertices, i.e a straight line between the nodes .In general, a street segment is defined as a poly-line however we can always divide it into a set of straight lines considering each corner of lines as a vertex to match our representation. Figure 3 illustrates the meaning of a street segment in $V^{'}$ as a convex combination. Φ denotes the

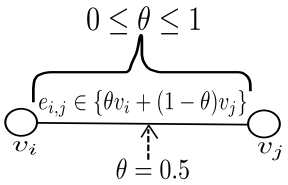


Figure 3: An edge in $G = (V,E)$ as a convex combination of two vertices

vector of mappings between a trajectory G and street network $G^{'}$. Unlike traditional graph matching where nodes of one graph are only matched to nodes of another graph, we need a mapping Φ which matches nodes of our input graph(trajectory) to any location over street segments in $E^{'}$: i.e. $\Phi_t = \Phi : v_t \mapsto e^{'} \times [0,1]$ where the interval $[0,1]$ specifies a value of θ pointing the exact location of mapping over a street segment in E'.
The truth value of vertex assignments to street segments can be stored in a binary matrix Δ, i.e. $\Delta \in \{0,1\}^{|V|\times|E^{'}|}$ subject to $\Delta^{\top}\mathbf{1} = \mathbf{1}$. Here $\mathbf{1}$ denotes a column vector of all ones. The last constraint enforces that each trajectory point can only be mapped to one street edge at a time.

Definition 3 *(Map Matching) Find a correspondence Φ between vertices of trajectory G and locations on street network G' such that the two matched sets, V and Φ, look most similar according to an objective criteria F_o. The problem is solved through finding an assignment of V to points lying on the edge set E' that minimizes the criteria F_o.*

The function F_o is of special interest and should have a form which preserves the relationships among input points while translating them to output space. We further assume that F_o is a decomposable function of a summation of basis functions, denoted by f_{Φ_i,Φ_j}.Thus to minimize F_o, we need to minimize the individual entries of summation. We also introduce the assignment matrix Δ in F_o. The assignment matrix Δ, enforces the Euclidean graph (or street network) constraints on the output i.e

$$F_o = \underset{\mathbf{\Phi}}{\operatorname{argmin}} \sum_{l}^{|E^{'}|} \sum_{i,j}^{|V|} \Delta_{i,l} \cdot f_{\Phi_i,\Phi_j}$$
$$s.t \;\; \Delta \in \{0,1\}^{|V|\times|E^{'}|}, \Delta^{\top}\mathbf{1} = \mathbf{1} \qquad (2)$$

Eq. (2) describes the map matching problem as a so-called integer linear program (ILP) which are generally known to be NP-hard except for a few classes of them. Such mathematical programs have a discrete and a continuous part. In our case, the discrete part chooses the correct combination of trajectory point ($v \in V$) versus street segment ($e' \in E'$). The total number of combinations is $V^{E'}$, which easily gets intractable even for a modest number of trajectory points and street network segments. Relaxation is a standard technique to reduce the complexity of ILP problems. In relaxation, we drop the discrete part of the problem to make it a standard linear programming problem where the optimization can be carried out in polynomial time. The result of the optimization is then rounded into a hard assignment

fulfilling the discrete constraints to get an approximate solution. In our case, discrete constraints amount to street network constraints and relaxation means ignoring these constraints. Eq. (3) describes the unconstrained F_o

$$F_o = \underset{\mathbf{\Psi}}{\operatorname{argmin}} \sum_{i,j}^{|V|} f_{\Psi_i,\Psi_j} \tag{3}$$

Notice that we have changed the output vector Φ to intermediate output Ψ. The mapping Ψ is different from Φ as it maps a a vertex v_t to $\mathbb{R}^2$: i.e $\Psi_t = \Psi : v_t \mapsto \mathbb{R}^2$.Now the solution of the optimization will be an approximation of the path used by trajectory instead of the original path. Afterwards we can convert this approximate path into the street network path through a rounding step. Hence, we provide a two step framework for the solution of Eq. (2)

- Optimize the relaxed objective function to approximate the trajectory path
- Provide a rounding scheme for assignment of step a output to street network.

We proceed by describing the details of these two steps in turn.

Optimization Step

Eq. (3) describes F_o as a summation of entries f_{Ψ_i,Ψ_j}. Now we come to details of an individual entry f_{Ψ_i,Ψ_j}. For this purpose, we define two Kernel matrices K_G and $K_{G'}$, where $K_{G_{i,j}}$ is the kernel function between trajectory vertices v_i and v_j: i.e $K_{G_{i,j}} = k_G(v_i, v_j)$ and $K_{G'_{i,j}}$ is the kernel function between the mappings Ψ_i, Ψ_j of the vertices v_i and v_j i.e $K_{G'_{i,j}} = k_{G'}(\Psi_i, \Psi_j)$. The widths of the Kernels K_G and $K_{G'}$ are denoted by σ_G and $\sigma_{G'}$. Now we define f_{Ψ_i,Ψ_j} as the 'difference of kernels' function.

$$f_{\Psi_i,\Psi_j} = (k_G(v_i, v_j) - k_{G'}(\Psi_i, \Psi_j))^2 \tag{4}$$

Now we can substitute the value of f_{Ψ_i,Ψ_j} in Eq. (3).

$$F_o = \underset{\mathbf{\Psi}}{\operatorname{argmin}} \sum_{i,j}^{|V|} (k_G(v_i, v_j) - k_{G'}(\Psi_i, \Psi_j))^2 \tag{5}$$

Eq. (5) comes from an embedding technique known as Multidimensional scaling described in Eq. (1). We further note that Eq. (5) is like a regression equation with multiple outputs where we want to preserve the correlation among inputs during our structured prediction process and it can be used for embeddings across different spaces and structures, however our case is a special case where the input and output spaces are the same. To encode our prior knowledge that the solution of the embedding lies in the spatial neighbourhood perturbed by Gaussian noise, we add a regularization term which fuses the input and output space into one. We propose a kernel function $k_{GG'}$ between the respective points of our input and output graphs and define it as

$$k_{GG'}(\lambda, \sigma_N, i) = e^{-((v_i - \Psi_i)^2 - \lambda\sigma_N^2)^2 / 2\sigma_N^2} \tag{6}$$

where λ is a stiffness parameter of the kernel function while σ_N , the kernel width, is the estimated standard deviation of noise in trajectory. We finalize our objective function F_o as following:

$$F_o = \underset{\mathbf{\Psi}}{\operatorname{argmin}} \sum_{ij} (K_{G_{i,j}} - K_{G'_{i,j}})^2 - \sum_i k_{GG'}(\lambda, \sigma_N, i) \tag{7}$$

Eq. (7) is the final objective function which we are using in kernelized map matching. However alternative embedding approaches can also be used in principal. We can take the derivative of F_o in Eq. (7) w.r.t Ψ and can find the answer. We use 'scaled conjugate gradient' algorithm for optimization [Shewchuk, 1994]. For assignment purposes, we apply a rounding scheme on the result of continuous optimization. We implement the optimization step in sliding window style of width k_w, because it makes the solution real time, localized and efficient.

Spatio-temporal Kernel over Euclidean Distances

The kernels $K_{G_{i,j}}$ and $K_{G'_{i,j}}$, that we have used are similar to RBF kernels. The only difference with an RBF kernel is that we use 'the sum of euclidean distances for all successive pairs of points between i and j' instead of using 'euclidean distance between i and j' directly. This little trick allows us to capture the spatial and temporal correlation among trajectory points in our kernel matrices. To elaborate further: for two points p_i and p_j in a trajectory where $j > i$, the sum of successive euclidean distances is denoted by $\delta_{i,j}$, and is defined as:

$$\delta_{i,j} = \sum_{i \leqslant m < j} |p_m, p_{m+1}|_2$$

The kernel $K_{i,j}$ is simply an RBF kernel with $\delta_{i,j}$ as the core part instead of the direct euclidean distance between i and j.

$$K_G(i,j) = e^{-\delta_{i,j}/\sigma_k} \tag{8}$$

A kernel is a valid kernel if there exists an embedding space for input points where it can be applied as a known kernel. One can show that our spatio-temporal kernel is a valid kernel because if we project a trajectory onto a straight line such that every successive distance is preserved and then take the RBF kernel over the points of this projection, it will be same as the above mentioned kernel. By virtue of being 'valid', the spatio-temporal kernel described can be integrated with other kernels in future to reflect more domain knowledge.

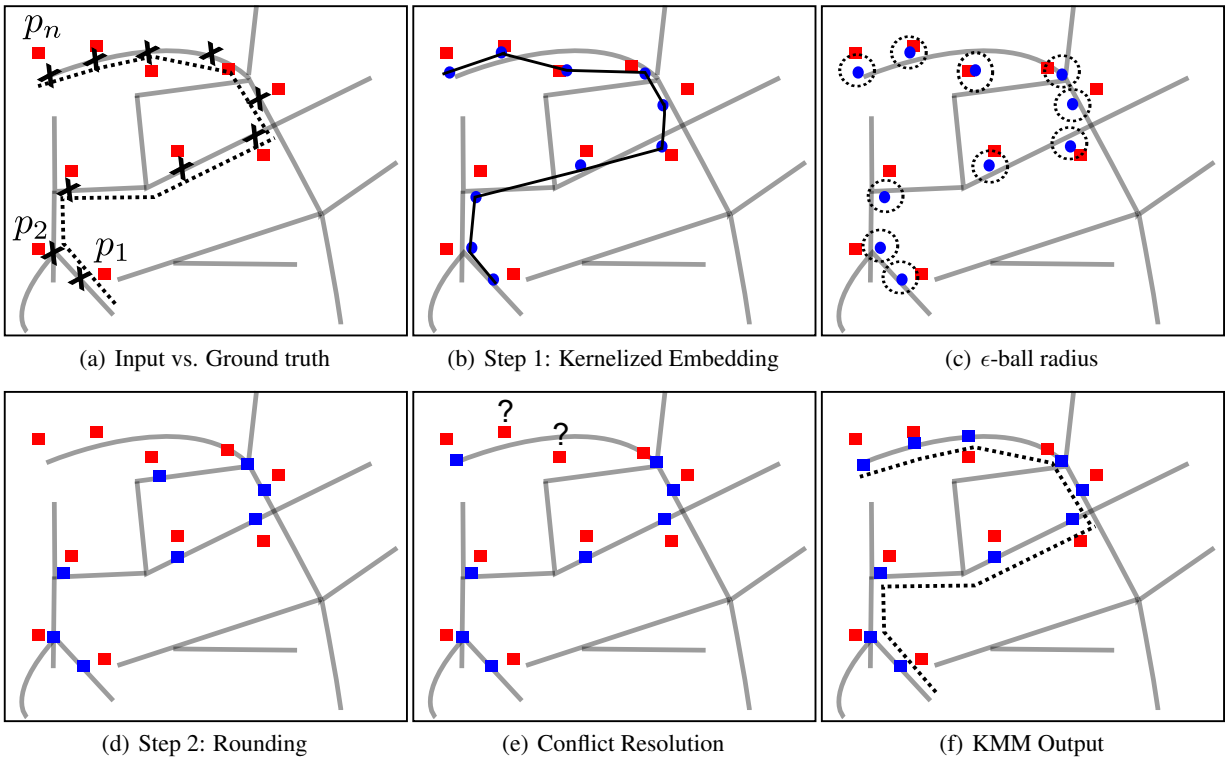
Rounding Step

After the noise is reduced by the optimization, we have to assign the points to street network. The simplest rounding scheme can be a nearest neighbour based one. However, we provide a more sensible scheme, which is based upon the following assumptions:

1. **Assumption 1**: For nearby points, the Euclidean distance between points resembles shortest street network graph distance;
2. **Assumption 2**: Most of the map matching algorithms use a radius ϵ to restrict the assignment possibilities.

For assigning a point Ψ_i to the street network, we pick all edges inside ϵ-neighbourhood and find nearest neighbours of Ψ_i on these edges. The resulting points are our candidates for the solution Φ_i. We denote the set of candidate solutions for Φ_i as C_{Φ_i}. According to our assumption 1, Ψ_i should be assigned to a point $c \in C_{\Phi_i}$ which minimizes the difference between euclidean distance and shortest path distance between the solution Φ_i and Φ_{i-1}. For this purpose, we take an RBF kernel K_Ω between graph distance (denoted by $\Omega(x, y)$) and Euclidean distance (denoted by $d(a, b)$) as following:

$$K_{\Omega_i} = e^{-(d(\Psi_i, \Psi_{i-1}) - \Omega(c \in C_{\Phi_i}, \Phi_{i-1}))^2} \tag{9}$$



(a) Input vs. Ground truth (b) Step 1: Kernelized Embedding (c) ϵ-ball radius

(d) Step 2: Rounding (e) Conflict Resolution (f) KMM Output

Figure 4: Step-by-step illustration of Kernelized Map Matching. (a) Map matching input with ground truth values. Red squares denote the input Trajectory; Gray edges, the street network graph; dashed lines, the ground truth path and crosses the ground truth points. (b) Approximation of ground-truth path by kernelized embedding in a relaxed setting. Blue circles denote the optimization output Ψ. (c) Imposition of $\epsilon-$ball radius to reduce the number of candidate matches for rounding. Dashed-circles denote $\epsilon-$balls. (d) Hard assignment of Ψ to street network graph in the rounding step on the basis of RBF kernels Blue squares denote the assignment. (e) A conflict and its resolution. The graph distance is greater than $\alpha\times$ Euclidean distance, both points are discarded to get a vote from neighbours. (f) The final output points and path computed by KMM. (Best viewed in color)

K_Ω stipulates the assumption 1 for two consecutive points. However to avoid K_Ω output going away from input point, we add a regularizer term which is also an RBF kernel between a Ψ_i and the candidate in question. Now for all elements $c \in C_{\Phi_i}$ we calculate the following term

$$-e^{-(c \in C_{\Phi_i} - \Psi_i)^2} \cdot K_{\Omega_i} \tag{10}$$

The candidate point which gives the minimum value for this term is chosen as solution Φ_i. In most of the cases, the comparison term between candidates, i.e. regularized K_{Ω_i} produces the right result however it is possible that after assignment the graph distance between Φ_i and Φ_{i-1} is far greater than the Euclidean distance between them, which introduces a conflict and violates our observation 1. To check these we take a constant α and multiply euclidean distance by it. After assignment, if the graph distance $\Omega(\Phi_i, \Phi_{i-1})$ is greater than $\alpha \times d(\Psi_i, \Psi_{i-1})$ (α-condition); we employ a resolution scheme, inspired by [Newson and Krumm, 2009]. We discard Φ_i and Φ_{i-1} and consider Φ_{i-2} as our previous point instead of Φ_i. After the assignment Φ_{i+1} , we again see whether the condition 1 is violated or not, if it happens again, we discard Φ_{i+1} and Φ_{i-2} and go ahead in the same fashion. We continue doing so until α-condition is not violated any more after an assignment of a point denoted by $\Phi_{\hat{i}}$. Once we find such a point $\Phi_{\hat{i}}$, we can again resume the standard comparison. However, before resuming the comparison, we assign all the discarded points on the shortest path between $\Phi_{\hat{i}}$ and the corresponding previous point. A conceptual work-flow of KMM steps together with explanations for each step are provided in Fig. 4

Parameter Selection

KMM has four input parameters, namely, σ_n, standard deviation of noise; λ, stiffness parameter for regularization term of F_o; α, the rounding step parameter and k_w, width of sliding window for objective function F_o.We discuss selection process for these parameter as following:
σ_N is required as an input to F_o in the regularization term. Our method for estimating σ_N is carried out on a holdout sample of 20 ground truth points and the approach is same as [Newson and Krumm, 2009]. Alternative strategies include estimators from other map matching solutions or empirical standards for given application. λ is the stiffness parameter of regularization term. A reasonable range for choice of λ is between 0.5 and 1.Clearly, $\lambda > 1$ make the standard deviation of Ψ more than the trajectory while $\lambda < 0.5$ is too restrictive on output. In our experiments, have chosen λ=0.6. k_w is the sliding window width for execution of optimization step. Figure 6 shows the results for $k_w = \{3, 4, .., 6\}$ over a synthetic data set. We have chosen $k_w = 5$ for most of our experiments. According to our observation, $k_w > 6$ proves a bit time consuming during optimization while $k_w < 3$ is not sufficient to capture the local geometry. α- is the parameter 'governing relationship between Graph and Euclidean distance'. We set it to 1.5 which means that Graph distance should not be greater than 1.5$\times$Euclidean distance. A reasonable range is $1 \leq \alpha \leq 2$. σ_{K_G} and $\sigma_{K_{G'}}$ are learned through optimization over a holdout sample of 20 consecutive input.

Algorithm 1: KMM

Input : G,G', K_G,$K_{G'}$,$K_{GG'}$,K_Ω,λ,k_w, α,σ_N
Output: $\Phi(V)$ - The Output Path P

// G, G'- Euclidean Graphs
// $K_G, K_{G'}, K_{GG'}, K_\Omega$ -Graph Kernels
// λ -Regularizer
// k_w- sliding window width
// α- constant for Euclidean versus Graph distance validation

for $i \leftarrow 1 : |V| - k_w$ **do**
  foreach *sliding window* **do**
    compute $K_G, K_{G'}$
    $\Psi \leftarrow$ Optimize F_o w.r.t Ψ

$i \leftarrow 1$, $i_{prev} \leftarrow 1$, $prev_{dist} \leftarrow 0$
while $i < |V|$ **do**
  $i \leftarrow i + 1$
  $i_{prev} \leftarrow max(1, i_{prev})$
  $min_{obj} \leftarrow \infty$
  if $i_{prev} = i - 1$ **then**
    $prev_{dist} \leftarrow 0$
    $con_{dist} \leftarrow d(\Psi_i, \Psi_{i-1})$
  else
    $con_{dist} \leftarrow$
    $d(\Psi_i, \Psi_{i-1}) + prev_{dist} + d(\Psi_{i_{prev}}, \Psi_{i_{prev}+1})$
  for $c \in C_{\Phi_i}$ **do**
    $obj_{val} \leftarrow -e^{-(c-\Psi_i)^2}.K_{\Omega_i}$
    if $obj_{val} < min_{obj}$ **then**
      $min_{obj} \leftarrow obj_{val}$
      $\Phi_i \leftarrow c$
  if $\Omega(\Phi_i, \Phi_{i_{prev}}) > \alpha \times d(\Psi_i, \Psi_{i_{prev}})$ **then**
    $i_{prev} \leftarrow i_{prev} - 1$
    $prev_{dist} \leftarrow con_{dist}$
  else
    Assign all points between i and i_{prev} to shortest path between Φ_i and $\Phi_{i_{prev}}$

Output path P by connecting consecutive Φ

4 Experimental Evaluation

In this section we report the results from a series of experiments which we conducted in order to empirically investigate the following questions:

(Q1) Can kernel methods be used for map matching?

(Q2) If so, how do they perform compared to state-of-the-art methods?

(Q3) Is kernelized embedding indeed reducing the noise?

(Q4) Does the rounding step in KMM contribute to the error reduction?

To this aim, we implemented KMM in scientific Python running on a standard Intel-Quadcore 2GHz computer.

Overall, we decided for two experimental setups. Our first experimental setup evaluates and compares KMM's accuracy performance on a real-world dataset recently used in [Newson and Krumm, 2009] to evaluate an hidden Markov model based map matching approach and hence addresses **Q1**, **Q2**. To address **Q4** we compare the performance of our rounding step with a randomized rounding step. The second setup investigates **Q3** by comparing the result of KMM's embedding step to baseline "closest point on edge" using synthetically generated dataset.

4.1 Q1+Q2+Q4: Real-World Dataset

In our first experiment, we compared KMM's performance to the performance of Krumm and Newson's recent hidden Markov model based approach [Newson and Krumm, 2009]. To measure performance, we used the Route Mismatch Fraction measure already used by Newson and Krumm. Route Mismatch Fraction(RMF) is the total length of false positive and false negative route divided by length of original route [Newson and Krumm, 2009]:

$$d_+ = \text{length of erroneously added route}$$
$$d_- = \text{length of erroneously subtracted route}$$
$$d_o = \text{length of original route}$$
$$RMF = \frac{(d_+ + d_-)}{d_o}$$

We used Krumm and Newson's dataset. It consists of a 50-mile route in Seattle sampled at 1 Hz, giving one trajectory of 7531 time stamped latitude/longitude pairs along with manually matched ground-truth path. The street network comprises around $150,000$ road segments. [Newson and Krumm, 2009] presented results for different sampling intervals and noise degradations of this data. We take six base cases where HMM model performance is good, i.e 5,10 seconds sampling intervals vs. 30,40,50 meters noise. Because we want to have a statistical comparison (average, significance,etc.) with HMM, we perform 25 experiments for comparison with each base value in the following way. For one sampling interval, say 10, we choose 5 different starting points from the initial 5 points of the trajectory and then sampled at the given rate. Following this procedure, we prepared experimental datasets for each sample by adding 5 instances of noise for one standard deviation (e.g 30), i.e

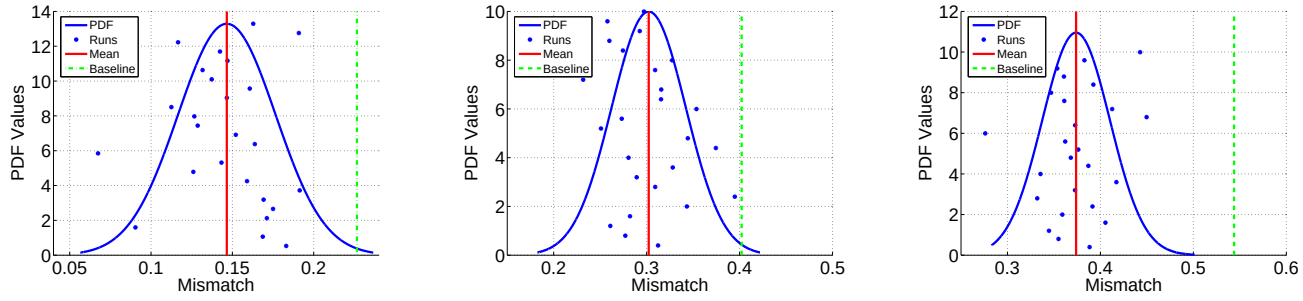
25 datasets/combination = 5 samples $\times$5 noise instances.

Fig. 5 summarizes the experimental results showing the RMF errors. As one can see, in 5 out of 6 cases KMM estimated a lower route mismatch fraction. The statistical significances of the results are shown in Table 1. In 4 out of 5 cases where we are better, the differences in mean values are significant (t-test, $p = 0.05$). Averaged over all experiments, a Wilcoxon test identifies KMM to be significantly better.
To address **Q4**, we compared the performance of our rounding step with a randomized rounding step, i.e random assignment of embedding output Ψ to a street network point in ϵ-ball, over the real-world dataset described above. Table 2 summarizes the results. As one can see our rounding step always performs better, and in most cases outperforms the randomized rounding step by a margin of 4-10 percent in error.

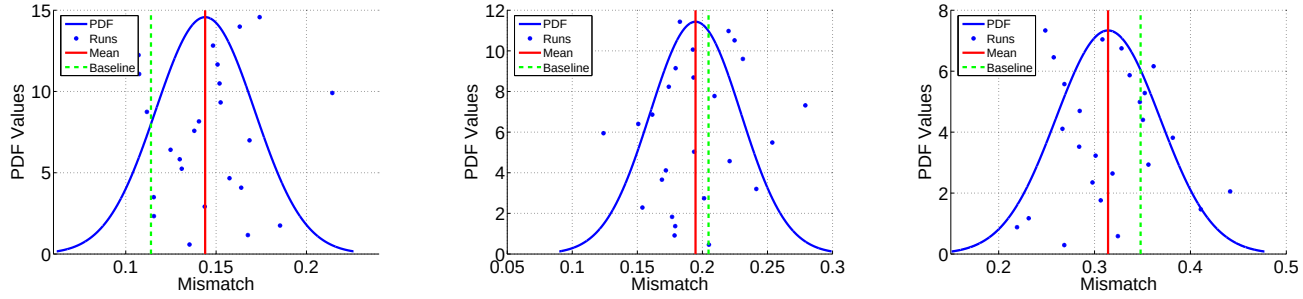
To summarize, the results clearly answer questions **Q1** ,**Q2** and **Q4** affirmatively.

4.2 Q3: Synthetic Datasets

In order to investigate how well the embedding step reduces noise, we generate ground truth points with the help of synthetic data. To generate the data, we used Thomas Brinkhoff's data generator [Brinkhoff, 2000]. It allows to generate trajectories according to some underlying road network for different speed and sampling time variation setting. Additionally, we assumed normal noise on the

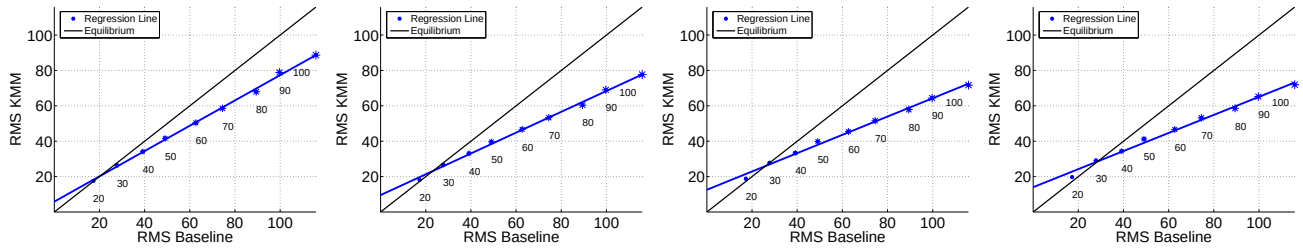


(a) NL 30, SR 5 (b) NL 40, SR 5 (c) NL 50, SR 5



(d) NL 30, SR 10 (e) NL 40, SR 10 (f) NL 50, SR 10

Figure 5: Route Mismatch Fraction of KMM vs. HMM for different noise level (NL, in meters) and sampling rates (SR, in seconds). We ran 150 experiments i.e 25 experiments/per NL-SR setting. One blue dot denotes the route mismatch fraction for an experiment, blue graphs the estimated normal distributions, red lines the means, and dashed green lines the route mismatch fraction for an HMM as reported by Newson and Krumm. As one can see, in 5 out of 6 cases KMM estimates a lower route mismatch fraction. In 4 out of 5 cases, the differences in mean values are significant (t-test, $p = 0.05$). Averaged over all experiments, a Wilcoxon test identifies KMM to be significantly better. (Best viewed in color)



(a) K=3 (b) K=4 (c) K=5 (d) K=6

Figure 6: Scatter plots of the root mean squared(RMS) error of KMM and of the baseline, closest point on edge, for different window sizes (K) on the Oldenburg dataset for different noise levels $(20, 30, 40, \ldots, 100)$ as denoted by the numbers associated with the dots. For better comparison with the equilibrium (solid straight line), we also show the linear regression line of the values. As one can see KMM performs much better with increasing noise levels. The regression lines have consistently a smaller slope than the "equilibrium" line. The turning point is around noise level 25. (Best viewed in color)

generated observations. it is well known navigation systems produce noise that is normally distributed with average of noise (σ_N) ranging from 10-100 meters. For instance, Krumm and Newson [Newson and Krumm, 2009] discussed the different types and amount of noise in real-world movement data and have shown that they can be described well by a Gaussian shape.

Specifically, we generated 100 trajectories of average length 50 from the street network of Oldenburg, Germany. Then, we added noise σ_N for $\sigma_N = 20, 30, \ldots, 100$. This resulted in an overall set of 900 trajectories. As baseline for comparison we used a "project to the closed point in the street network" approach. To get a fair comparison, we also used the "project to the closed point in the street network" as "rounding" method for KMM. We report on the root means squared difference in meters achieved by the two methods.

The results are summarized in Fig. 6. As one can see, in all cases the embedding indeed reduced noise considerably. Moreover, it performs better with increasing noise levels as the regression lines have consistently a smaller slope than the "equilibrium" line. The turning point is around noise level 25. This clearly answers question **Q3** affirmatively. To summarize, the results of our experiments indicate that kernel methods can indeed be used for map matching and achieve results comparable to state-of-the-art techniques.

5 Conclusion

In this paper, we have recognized that kernel methods have an important role to play in map matching. Specifically, we have established — to the best of our knowledge — the first link between map matching and kernel methods and instan-

Statistics for sampling rate=5			
Noise	σ_{error}	student t	Wilcoxon signed rank
30	0.029	1	0.00012
40	0.03992	1	0.0027
50	0.036	1	0.00122
Statistics for sampling rate=10			
Noise	σ_{error}	student t	Wilcoxon signed rank
30	0.027	1	0.0004
40	0.034	0	0.047
50	0.054	1	0.00941

Table 1: Statistics table for comparison with HMM

Noise Ratio	KMM Error	ERR SR=5	ERR SR=10
30	0.1463	0.1783	0.1480
40	0.3022	0.3221	0.2350
50	0.3739	0.4671	0.3979

Table 2: Comparison of average KMM error with average Randomized Rounding (ERR) error over different Noise Ratio and Sampling Rates (SR)

tiated the link by developing an easy-to-implement kernelized map matching (KMM) approach. By accounting for spatial and temporal correlations among the trajectories using kernels, map matching becomes less sensitive to noisy observations. This allowed us to employ a simple rounding scheme to compute the hard assignments of trajectory points to points lying on the network. Our experimental results on both simulated as well as real-world datasets show that kernel methods can indeed be used for map matching and can achieve better performance than state-of-the-art hidden Markov models.

The link established between map matching and kernels provides many interesting avenues for future work; we have only started to explore it. Indeed, one should study alternative kernels and map features, cost functions and hyper-parameter optimization criteria. For instance, so-called Fisher kernels are kernels derived from hidden Markov models. In turn, we may utilize any HMM based map matching approach. Testing KMM within a real-world system tracking system is another interesting avenue. Proving the hardness of map matching problems along with guarantees on approximation are interesting venues of future work. Finally, KMM directly generalize to the case of 3D trajectories.

Acknowledgements

The authors would like to thank Paul Newson and John Krumm for making their data available, to anonymous reviewers for helpful suggestions and Novi Quadrianto for taming the notation and ideas in this paper . This work was supported by the Fraunhofer ATTRACT fellowship STREAM and a "Higher Education Commission" (HEC)-Pakistan scholarship.

References

[Brakatsoulas *et al.*, 2005] Sotiris Brakatsoulas, Dieter Pfoser, Randall Salas, and Carola Wenk. On map-matching vehicle tracking data. In *VLDB*, pages 853–864. VLDB Endowment, 2005.

[Brinkhoff, 2000] Thomas Brinkhoff. Generating network-based moving objects. In *SSDBM*, page 253, Washington, DC, USA, 2000. IEEE Computer Society.

[Cox and Cox, 2000] Trevor F. Cox and M.A.A. Cox. *Multidimensional Scaling, Second Edition*. Chapman and Hall/CRC, 2 edition, 2000.

[Fox *et al.*, 2003] Dieter Fox, Jeffrey Hightower, Lin Liao, Dirk Schulz, and Gaetano Borriello. Bayesian filtering for location estimation. *IEEE Pervasive Computing*, 2(3):24–33, 2003.

[Guo *et al.*, 2008] Y. Guo, J. Gao, and P.W. Kwan. Twin kernel embedding. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(8):1490–1495, 2008.

[Lee and Verleysen, 2007] John A. Lee and Michel. Verleysen. *Nonlinear dimensionality reduction*. Springer, New York; London, 2007.

[Lou *et al.*, 2009] Yin Lou, Chengyang Zhang, Yu Zheng, Xing Xie, Wei Wang, and Yan Huang. Map-matching for low-sampling-rate gps trajectories. In *GIS*, pages 352–361, New York, NY, USA, 2009. ACM.

[Mitchell, 2009] Tom Mitchell. Mining our reality. *Science*, 326(5960):1644–1645, 2009.

[Newson and Krumm, 2009] Paul Newson and John Krumm. Hidden markov map matching through noise and sparseness. In *GIS*, pages 336–343, New York, NY, USA, 2009. ACM.

[Pemmaraju and Skiena, 2003] Sriram Pemmaraju and Steven Skiena. *Computational Discrete Mathematics: Combinatorics and Graph Theory with Mathematica* . Cambridge University Press, 2003.

[Quadrianto *et al.*, 2009] Novi Quadrianto, Le Song, and Alex J. Smola. Kernelized sorring. In *NIPS 21*, pages 1289–1296. 2009.

[Quddus *et al.*, 2007] Mohammed A. Quddus, Washington Y. Ochieng, and Robert B. Noland. Current map-matching algorithms for transport applications: State-of-the art and future research directions. *Transportation Research Part C: Emerging Technologies*, 15(5):312 – 328, 2007.

[Shawe-Taylor and Cristianini, 2004] John Shawe-Taylor and Nello Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, illustrated edition edition, 2004.

[Shewchuk, 1994] Jonathan R Shewchuk. An introduction to the conjugate gradient method without the agonizing pain. Technical report, Pittsburgh, PA, USA, 1994.

[Taylor *et al.*, 2001] George Taylor, Geoffrey Blewitt, Doerte Steup, Simon Corbett, and Adrijana Car. Road reduction filtering for gps-gis navigation. *Transactions in GIS*, 5(3):193–207, 2001.

[Weston and Vapnik, 2002] Chapelle O. Elisseeff A. Scholkopf B. Weston, J. and V. Vapnik. Kernel dependency estimation. *Advances in neural information processing systems*, 2002.

Convex NMF on Non-Convex Massiv Data

Kristian Kersting[1] and **Mirwaes Wahabzada**[1] and **Christian Thurau**[2] and **Christian Bauckhage**[2]
[1]Knowledge Discovery Department, [2]Vision and Social Media Group
Fraunhofer IAIS, Schloss Birlinghoven, 53754 Sankt Augustin, Germany
firstname.lastname@iais.fraunhofer.de

Abstract

We present an extension of convex-hull non-negative matrix factorization (CH-NMF) which was recently proposed as a large scale variant of convex non-negative matrix factorization (C-NMF) or Archetypal Analysis (AA). CH-NMF factorizes a non-negative data matrix V into two non-negative matrix factors $V \approx WH$ such that the columns of W are convex combinations of certain data points so that they are readily interpretable to data analysts. There is, however, no free lunch: imposing convexity constraints on W typically prevents adaptation to intrinsic, low dimensional structures in the data. Alas, in cases where the data is distributed in a non-convex manner or consists of mixtures of lower dimensional convex distributions, the cluster representatives obtained from CH-NMF will be less meaningful. In this paper, we present a hierarchical CH-NMF that automatically adapts to internal structures of a dataset, hence it yields meaningful and interpretable clusters for non-convex datasets. This is also conformed by our extensive evaluation on DBLP publication records of 760,000 authors, 4,000,000 images harvested from the web, and 150,000,000 votes on World of Warcraft guilds.

1 Introduction

Modern applications of data mining and machine learning in computer vision, natural language processing, computational biology, and other areas often consider massive datasets and we need to run expensive algorithms such as principle component analysis (PCA), latent Dirichlet allocation (LDA), or non-negative matrix factorization (NMF) to extract meaningful, low-dimensional representations.

If the data are words contained in documents, these methods yield topic models representing each document as a mixture of a small number of topics and each word is attributable to one of the topics. In computer vision, where it is common to represent images as vectors in a high-dimensional space, they extract visual words and have been used for face and object recognition, or color segmentation. Social networks such as Flickr, Facebook, or Myspace allow for a wide range of interactions amongst their members, resulting in massive, temporal datasets relating users, media objects, and actions. Here, low-dimensional representations may identify and summarize common social activities.

Therefore, given massive matrices of hundreds of millions of entries, how can we *efficiently* factorize them? How can we create *meaningful*, low-dimensional representations? How can we gain inside into the dataset? These are precisely the questions which we address in this paper.

A recent positive development in data mining and machine learning has been the realization that massive datasets are not only challenging but may as well be viewed as an opportunity [Torralba *et al.*, 2008; Talwalkar *et al.*, 2008]. Machine learning and data mining techniques typically consist of two parts: the model and the data. Most effort in recent years has gone into the modeling part. Massive datasets, however, allow one to move into the opposite direction: *how much can the data itself help us to solve the problem*? Halevy *et al.* [2009] even speak of the "*the unreasonable effectiveness of data*". Massive datasets are likely to capture even very rare aspects of the problem at hand. Along this line, Thurau *et al.* [2009] have recently introduced a data-driven NMF approach, called convex-hull NMF, that is fast and scales extremely well: it can efficiently factorize gigantic matrices and in turn extract meaningful "clusters" from massive datasets containing millions of images and ratings. The key idea is to restrict the "clusters" to be combinations of vertices of the convex hull of the dataset; thus directly exploring the data itself to solve the convex NMF problem.

There is, however, no free lunch: by restricting the "clusters" to combinations of vertices of the convex hull of the dataset, convex-hull NMF cannot adapt to the intrinsic, low dimensional structure of the data anymore. Intuitively, for data modeled by Gaussians, i.e., as a combination of convex sets, convex-hull NMF will assign clusters to the "extreme" Gaussians and not to each Gaussian. Our main contribution is a simple and, hence, powerful and scalable generalization of convex hull NMF that automatically adapts to the intrinsic (low dimensional) structure in the data. The main insight is that one can use FastMap due to Faloutsos and Lin [1995] within convex-hull NMF to compute the convex hull vertices on massive datasets. Consequently, we can still solve the convex NMF problem but additionally adapt to the intrinsic structure in the data, when we split the data along each 1D FastMap line recursively, stop when some minimum node size is reached, and apply a post-pruning step. This way, we get meaningful clusters that respect the structure of the data, i.e., that diversify the results even better than convex-hull NMF. Our extensive experimental evaluation shows that the method, called hierarchical convex-hull NMF, achieves similar reconstruction quality as convex-hull NMF with only a small overhead while it produces more diversified clusters on DBLP publication records of 760.000 authors, 4 million tiny im-

(a) NMF **(b)** C-NMF **(c)** CH-NMF **(d)** HCH-NMF

Figure 1: The basis vectors resulting from different NMF variants applied to the CBCL Face Database 1. (a) Standard NMF results in part-based, sparse representations. Data points cannot be expressed as convex combinations of these basis elements. (b) Convex NMF (C-NMF) yields basis elements that allow for convex combinations. Moreover, the basis vectors are "meaningful" since they closely resemble given data points. They are, however, not indicative of characteristic variations among individual samples. (c) Convex-Hull NMF (CH-NMF) diversifies the basis vectors resulting in pale faces, faces with glasses, faces with beards, and so on. (d) Hierarchical CH-NMF (HCH-NMF) as proposed in the current paper also diversifies the results. Additionally, it automatically groups them, i.e., it identifies structure within the data. The induced hierarchical decomposition is shown together with the number of images falling into the corresponding subtrees.

ages, and 150 million votes on World of Warcraft guilds.

We proceed as follows. We start off by briefly reviewing non-negative matrix factorization (NMF) in Section 2, including convex NMF and convex-hull NMF. Then, we introduce hierarchical convex-hull NMF in Section 3. Before concluding, we present our extensive experimental evaluation in Section 4.

2 Non-Negative Matrix Factorization

Assume an $m \times n$ input data matrix $\mathbf{V} = (\mathbf{v_1}, \ldots, \mathbf{v_n})$ consisting of n column vectors of dimensionality m. We consider factorizations of the form $\mathbf{V} \approx \mathbf{W}^{m \times k}\mathbf{H}^{k \times n}$. The resulting matrix $\mathbf{W}$ contains a set of $k \ll n$ basis vectors which are linearly combined using the coefficients in $\mathbf{H}$ to represent the data. Common approaches to achieve such a factorization include Principal Component Analysis (PCA) [Jolliffe, 1986], Singular Value Decomposition (SVD) [Golub and van Loan, 1996], Vector Quantization (VQ), or non-negative Matrix Factorization (NMF) [Lee and Seung, 1999].

Various variants and improvements to NMF have been introduced in recent years. For example, Cai *et al.* [2008] presented a matrix factorization that obeys the geometric data structure. In [Kim and Park, 2008], a speed improvement to NMF is achieved using a novel algorithm based on an alternating nonnegative least squares framework. Another interesting variation is presented in [Suvrit, 2008] where optimization is based on a block-iterative acceleration technique. Recently, Mairal *et al.* [2010] have presented a very elegant online NMF approach based on sparse coding that also scales to large matrices (for additional related work, please see reference in [Mairal *et al.*, 2010]). In this work, however, we build on *Convex-NMF (C-NMF)* recently introduced by Ding *et. al* [2009], and it is not clear how to adapt these advanced NMF techniques to it as Convex-NMF represents the data matrix $\mathbf{V}$ as a convex combination of data points, i.e. $\mathbf{V} = \mathbf{VGH}^T$ where each column i of $\mathbf{G}$ is a stochastic vector that obeys $\|\mathbf{g}_i\|_1 = 1, \mathbf{g}_i \geq \mathbf{0}$. This is akin to *Archetypal Analysis* according to Cutler and Breiman [1994] where both matrices $\mathbf{G}$ and $\mathbf{H}^T$ are to be stochastic. Convex-NMF yields interesting interpretations of the data because each data point is now expressed as a weighted sum of certain data points (see Fig. 1).

Convex NMF Convex non-negative matrix factorization (C-NMF) was introduced by Ding *et al.* [2009] and minimizes $J = \|\mathbf{V} - \mathbf{VGH}^T\|^2$, where $\mathbf{V} \in \mathbb{R}^{m \times n}, \mathbf{G} \in \mathbb{R}^{n \times k}, \mathbf{H} \in \mathbb{R}^{n \times k}$. The matrices $\mathbf{G}$ and $\mathbf{H}$ are updated iteratively until convergence using the following update rules

$$G_{ik} = G_{ik}\sqrt{\frac{(\mathbf{Y}^+\mathbf{H})_{ik} + (\mathbf{Y}^-\mathbf{GH}^T\mathbf{H})_{ik}}{(\mathbf{Y}^-\mathbf{H})_{ik} + (\mathbf{Y}^+\mathbf{GH}^T\mathbf{H})_{ik}}} \quad (1)$$

and

$$H_{ik} = H_{ik}\sqrt{\frac{(\mathbf{Y}^+\mathbf{G})_{ik} + (\mathbf{HG}^T\mathbf{Y}^-\mathbf{G})_{ik}}{(\mathbf{Y}^-\mathbf{G})_{ik} + (\mathbf{HG}^T\mathbf{Y}^+\mathbf{G})_{ik}}} \quad (2)$$

where $\mathbf{Y} = \mathbf{V}^T\mathbf{V}$, and the matrices $\mathbf{Y}^+$ and $\mathbf{Y}^-$ are given by $Y^+_{ik} = \frac{1}{2}|Y_{ik}| + Y_{ik}$ and $Y^-_{ik} = \frac{1}{2}|Y_{ik}| - Y_{ik}$, respectively.

For the initialization of $\mathbf{G}$ and $\mathbf{H}$ two methods are proposed. The first initializes to (almost) unary representations based on a k-means clustering of $\mathbf{V}$. The second assumes a given NMF or Semi-NMF solution. For further details on the algorithm and its initializations we refer to [Ding *et al.*, 2009].

Recall, however, that our goal is to analyse massive, high-dimensional datasets. Unfortunately, the C-NMF update rules (1) and (2) have a time complexity of $O(n^2)$. Moreover, although the iterative algorithm comes down to simple matrix multiplications, the size of the involved matrices quickly becomes another limiting factor (similar to the intermediate blowup problem in tensor decomposition [Kolda and Sun, 2008]), since $\mathbf{V}^T\mathbf{V}$ results in an $n \times n$ matrix. Switching to an online update rule would avoid memory issues but it would at the same time introduce additional computational overhead. Overall, we can say that C-NMF does not scale to large datasets. In the following, we will review *Convex-Hull NMF (CH-NMF)*, which is a recent C-NMF method that is well suited for for large-scale data analysis.

Convex-Hull NMF Convex-Hull NMF aims at a data factorization based on the data points residing on the data convex hull. Such a data reconstruction has two interesting properties: first, the basis vectors are real data points and mark, unlike in most other clustering/factorization techniques, the most extreme and not the most common data points. Second, any data point can be expressed as a convex and meaningful combination of these basis vectors. This

Algorithm 1: CH-NMF

Input: Data matrix $\mathbf{V}^{m\times n}$
Output: Matrices $\mathbf{X}$ and $\mathbf{H}$

1 Compute k eigenvectors $\mathbf{e}_l$, $l = 1\ldots k$ of the covariance matrix of $\mathbf{V}^{m\times n}$;
2 Project $\mathbf{V}$ onto the 2D-subspaces $\mathbf{E}_{o,q}^{2\times n} = \mathbf{V}^T[\mathbf{e}_o, \mathbf{e}_q]$, $o = 1\ldots k$, $q = 1\ldots k, o \neq q$;
3 Compute and mark convex hull data points $H_{cvx}(\mathbf{E}_{o,q})$ for each 2D projection;
4 Combine marked convex hull data points (using the original data dimensionality m) $\mathbf{S}^{m\times p} = \{H_{cvx}(\mathbf{E}_{1,2}), \ldots, H_{cvx}(\mathbf{E}_{k-1,k})\}$;
5 Optimize $J_S = \|\mathbf{S} - \mathbf{S}\mathbf{I}^{p\times k}\mathbf{J}^{k\times p}\|^2$ such that $\|\mathbf{i}_i\|_1 = 1$ for $\mathbf{i}_i \geq \mathbf{0}$ and $\|\mathbf{j}_i\|_1 = 1$ for $\mathbf{j}_i \geq \mathbf{0}$;
6 Optimize $J = \|\mathbf{v}_i - \mathbf{X}\mathbf{h}_i^T\|^2$ for $i = 1\ldots n$ where $\mathbf{X} = \mathbf{S}^{m\times p}\mathbf{I}^{p\times k}$ such that $\|\mathbf{h}_i\|_1 = 1$ and $\mathbf{h}_i \geq \mathbf{0}$;

offers interesting new opportunities for data interpretation as indicated in Fig. 1 and demonstrated in [Thurau *et al.*, 2009].

More precisely, following Ding *et al.* [2009], one seeks a factorization of the form $\mathbf{V} = \mathbf{V}\mathbf{G}\mathbf{H}^T$, where $\mathbf{V} \in \mathbb{R}^{m\times n}, \mathbf{G} \in \mathbb{R}^{n\times k}, \mathbf{H} \in \mathbb{R}^{n\times k}$. One further restricts the columns of $\mathbf{G}$ and $\mathbf{H}$ to convexity, i.e., $\|\mathbf{g}_i\|_1 = 1, \mathbf{g}_i \geq \mathbf{0}$ and $\|\mathbf{h}_j\|_1 = 1, \mathbf{h}_j \geq \mathbf{0}$. Indeed, Ding *et al.* [2009] also consider convex combinations but not for the matrix $\mathbf{H}$. In other words, CH-NMF aims at factorizing the data such that each data point is expressed as a convex combination of convex combinations of specific data points. The task now is to minimize

$$J = \|\mathbf{V} - \mathbf{V}\mathbf{G}\mathbf{H}^T\|^2 \tag{3}$$

such that $\|\mathbf{g}_i\|_1 = 1, \mathbf{g}_i \geq \mathbf{0}$ and $\|\mathbf{h}_j\|_1 = 1, \mathbf{h}_j \geq \mathbf{0}$. To do so, one sets $\mathbf{X} = \mathbf{V}^{d\times n}\mathbf{G}^{n\times k}$. The intuition is as follows. Since we assume a convex combination for $\mathbf{X}$, and by definition of the convex hull, the convex hull $H_{cvx}(\mathbf{V})$ of $\mathbf{V}$ must contain $\mathbf{X}$. Obviously, we could achieve a perfect reconstruction, giving $J = 0$, by setting $\mathbf{G}$ so that it would contain exactly one entry equal to 1 for each convex hull data point while all other entries were set to zero. Or more informal: *following the definition of the convex hull we can perfectly reconstruct any data point by a convex combination of convex hull data points.* Therefore, our goal becomes to solve Eq. (3) by finding k appropriate data points on the convex hull:

$$J = \|\mathbf{V} - \mathbf{X}\mathbf{H}^T\|^2 \tag{4}$$

such that $\mathbf{x}_i \in H_{cvx}(\mathbf{V}),\ i = 1, \ldots, k$.

Finding a solution to Eq. (4), however, is not necessarily straight forward. It is known that the worst case complexity for computing the convex hull of n points in m dimensions is $\Theta(n^{\frac{m}{2}})$. Moreover, the number of convex hull data points may tend to n for high dimensional spaces, see e.g. [Donoho and Tanner, 2005; Hall *et al.*, 2005] so that computing the convex hull of large data-sets quickly becomes practically infeasible. CH-NMF therefore seeks an approximate solution by subsampling the convex hull. It exploits the fact that any data point on the convex hull of a linear lower dimensional projection of the data also resides on the convex hull in the original data dimension. Since $\mathbf{V}$ contains finitely many points and therefore forms a polytope in $\mathbb{R}^m$, we can resort to the *main theorem of polytope theory*, see e.g. [Ziegler, 1995]. In our context, it says that every vertex of an affine image of P, i.e., every point of the convex hull of the image of P, corresponds to a vertex of P. Therefore computing the convex hull of several 2D affine projections of the data offers a way of subsampling $H_{cvx}(\mathbf{V})$. This is an efficient way as computing the convex hull of a set of 2D points can be done in $O(n \log n)$ time, [de Berg *et al.*, 2000].

This subsampling strategy is the main idea underlying CH-NMF and, indeed, various methods can be used for linearly projecting the data to a 2D space. Thurau *et al.* [2009] proposed to use PCA, i.e., projecting the data using pairwise combinations of the first d eigenvectors of the covariance matrix of $\mathbf{V}$ as summarized in Alg. 1. For massive, high-dimensional data, however, computing the covariance matrix may take a lot of time. Therefore, we propose to use instead Faloutsos and Lin's FastMap [1995], but please see next section.

The main point here is, triggered by the idea that the expected size of the convex hull of n Gaussian data points in the plane is $\Omega(\sqrt{\log n})$ [Hueter, 1999], CH-NMF extracts only approximately $p = j\sqrt{\log n}$ candidate points. This candidate set grows much slower than n. Given a candidate set of p convex hull data points $\mathbf{S} \in H_{cvx}(\mathbf{V})$, we now select those k convex hull data points that yield the best reconstruction of the remaining subset $\mathbf{S}$. This, again, can be formulated as a convex NMF optimization problem. We now have to minimize the following reconstruction error

$$J_S = \|\mathbf{S}^{m\times p} - \mathbf{S}^{m\times p}\mathbf{I}^{p\times k}\mathbf{J}^{k\times p}\|^2 \tag{5}$$

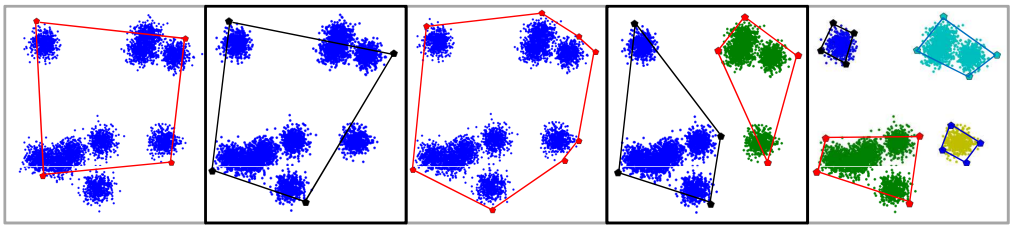
under the convexity constraints $\|\mathbf{I}_i\|_1 = 1, \mathbf{I}_i \geq \mathbf{0}$ and $\|\mathbf{J}_i\|_1 = 1, \mathbf{J}_i \geq \mathbf{0}$. Since $p \ll n$, solving (5) can be done efficiently using a quadratic programming solver. Note that the data dimensionality is m. The convex hull projection only served to determine a candidate set; all further computations are carried out in the original data space.

By obtaining a sufficient reconstruction accuracy for $\mathbf{S}$, we can set $\mathbf{X} = \mathbf{S}^{m\times p}\mathbf{I}^{p\times k}$ and thereby select k convex hull data points for solving Eq. (4). Typically, $\mathbf{I}$ results in unary representations. If this is not the case, we simply map $\mathbf{S}\mathbf{I}$ to their nearest neighboring data point in $\mathbf{S}$.

Given $\mathbf{X}$, the computation of the coefficients $\mathbf{H}$ is straight forward. For smaller data-sets it is possible to use the iterative update rule from Eq. (2). However, since we do not further modify the basis vectors $\mathbf{X}$, we can also find an optimal solution for each data point $\mathbf{v}_i$ individually $J_i = \|\mathbf{v}_i - \mathbf{X}\mathbf{h}_i\|^2$ using common solvers. Obviously, this can be parallelized.

3 Hierarchical CH-NMF

Modern datasets are not only massive, but often very complex. This makes it challenging to find useful information in the data. Fortunately, most datasets typically have a low intrinsic dimension. That is, the data lay on a smooth, structured low-dimensional manifold. As an illustrative (already low-dimensional) example consider Fig. 2. It depicts a typical "non-convex data" situation. We have drawn data points from 10 randomly positioned Gaussians in $2D$ and were interested in computing overcomplete representations, i.e., the number of basis vectors is greater than the dimensionality of the input. Overcomplete representations have been advocated because they have greater robustness in the presence of noise, can be sparser, and can have greater flexibility in matching structure in the data. By design, CH-NMF assigns clusters to the "extreme" Gaussians and not to "inner" Gaussians, cf. Figs. 2 **(a,c)**. Although we can still reconstruct each data point perfectly, this is discouraging. The intrinsic (low dimensional) structure of the data is not captured and, in turn, the representation of the data found is not as meaningful as it could be.



(a) CH-NMF bv=4 **(b)** HCH-NMF bv=4 **(c)** CH-NMF bv=8 **(d)** HCH-NMF bv=8 **(e)** HCH-NMF bv=16

Figure 2: Resulting basis vectors of CH-NMF **(a,c)** for 4 resp. 8 basis vectors (bv) and of HCH-NMF **(b,d,e)** for 4, 8, resp. 16 basis vectors. The data samples are drawn from 10 randomly placed Gaussian distributions in $2D$, 500 samples per Gaussian. For 4 basis vectors **(b)**, HCH-NMF essentially mimics CH-NMF. For more basis vectors **(d,e)**, however, it starts to adapt to the structure of the data: the basis vectors reside on the convex hulls of the Gaussians. CH-NMF's basis vectors **(a,c)**, in contrast, remain residing on the convex hull. By design, it considers the "extreme" Gaussians only and does not adapt to the structure of the data. (Best viewed in color.)

Algorithm 2: HCH-NMF: Hierarchical Convex-Hull NMF

Input: Data, i.e., the set of rows $\mathbf{v}_i$ of $\mathbf{V}^{m \times n}$; the pairwise distances $\mathbf{D}$; the minimal size *MinSize* of a leave
Output: A hierarchical decomposition of D represented as tree T

1 **if** $|\mathbf{V}| <$ *MinSize* **then**
2 **return** CH-NMF(*Leaf*) for l (even multiple of k) basis vectors
3 **else**
4 *Rule* $\leftarrow$CHOOSERULE($\mathbf{V}$);
5 $\mathbf{V}_s \leftarrow \{\mathbf{v}_i \in \mathbf{V} \mid \mathit{Rule}(\mathbf{v}_i) = \mathit{true}\}$;
6 $\mathbf{V}_f \leftarrow \{\mathbf{v}_i \in \mathbf{V} \mid \mathit{Rule}(\mathbf{v}_i) = \mathit{false}\}$;
7 *LeftTree* $\leftarrow$MAKETREE($\mathbf{V}_s$);
8 *RightTree* $\leftarrow$MAKETREE($\mathbf{V}_f$);
9 **return** (*Rule*, *LeftTree*, *RightTree*)

Hierarchical convex-hull NMF (HCH-NMF) is a convex NMF approach that automatically adapts to the low intrinsic dimensionality of data as illustrated in Figs. 2 **(b,d,e)**. The elegance of HCH-NMF stems from two facts:

- It naturally falls out of running CH-NMF using FastMap [Faloutsos and Lin, 1995] for efficiently computing and marking convex hull data points.
- In turn, it provably solves the convex NMF problem as it directly makes CH-NMF manifold-adaptive.

The latter point is difficult to prove for a two-steps approach: run any clustering approach, then run CH-NMF on the clusters. Also employing any of the existing large-scale manifold learning methods, see e.g. [Talwalkar *et al.*, 2008] is difficult. As Talwakar *et al.* [2008], argue they require a $\mathcal{O}(n^3)$ spectral decomposition of matrices where n is the number of samples. When the matrix is sparse, these techniques can be implemented relatively efficiently. However, when dealing with a large, dense matrix, the involved matrix products become expensive to compute.

As summarized in Alg. 2, HCH-NMF is based on a hierarchical decomposition of $\mathbb{R}^{\mathbb{D}}$ in form of a tree[1]. That is, it starts with the empty tree and repeatedly searches for the best test for a node according to some splitting criterion such as weighted variance along the FastMap dimension. Next, the examples $\mathbf{V}$ in the node are split into $\mathbf{V}_s$ (success) and $\mathbf{V}_f$ (failure) according to the test. For each split, the procedure is recursively applied, obtaining subtrees for the respective splits. We stop splitting if a minimum number *MinSize* of examples is reached or the variance in one node is small enough. In the leaves, we run CH-NMF on the examples falling into the leaves to find l basis vectors. Finally, we may run a post-processing step to find the best k basis vectors. In other words, HCH-NMF is conceptually easy, yet scalable to massive datasets and powerful as our experimental results will demonstrate.

Let us briefly review FastMap. FastMap computes a u-dimensional Euclidean embedding and proceeds as follows. Given pairwise distances among objects, in our case the rows of the data matrix $\mathbf{V}$, we select a pair of distant objects called *pivot objects*. Then, we draw a line between the pivot objects. Essentially, it serves as the first coordinate axis. For each object o, we determine the coordinate value $fm(o)$ along this axis by projecting o onto this line. Next, the pairwise distances of all objects are updated to reflect this projection, i.e., we compute the pairwise distances among the objects in the subspace orthogonal to the line. This process is repeated until, after u iterations, we get the u coordinates as well as the u-dimensional representation of all objects. Ostrouchov and Samatova [2005] have shown that the pivots are taken from the faces, usually vertices, of the convex hull of the data points in the original implicit Euclidean space. This justifies the idea to employ FastMap in step (3) "compute and mark convex hull data points" of CH-NMF as already mentioned in the last section. More important for HCH-NMF, it suggests a natural splitting criterion to produce a hierarchical decomposition: *Split the data according to the weighted variance along the 1D FastMap line.* More precisely, we compute the splitting rule as summarized in Alg. 3. Essentially, we run one iteration of FastMap and split along the "FastMap" line w.r.t. weighted variance. That is, we pick a pair of distant objects $\mathbf{x}$ and $\mathbf{y}$ (lines 1-3). Then, we project the data onto the line and compute for each data point its 1D coordinate value (lines 4-6). Now, we compute the split variable θ that minimizes the weighted variance (lines 8-12) and return the corresponding splitting rule (lines 13-14). Thus, for each splitting variable, determining the split point s can be done very quickly. In turn, by scanning all of the inputs, determining the best split is feasible and scales as $\mathcal{O}(n \log n)$.

How large should we grow the tree? Clearly, a very large tree might produce too many basis vectors. The basis vectors are likely to be not meaningful and the tree essentially "overfits" the data. A small tree, however, might not cap-

[1] In this paper, we follow Dasgupta and Freund's random projection trees RPTs) [Dasgupta and Freund, 2009a]. We do not split along a random direction but note that this could easily be achieved.

Algorithm 3: CHOOSERULE based on FastMap

Input: Data, i.e., the set of rows $\mathbf{v}_i$ of $\mathbf{V}^{m \times n}$; the minimal size of a leave *MinSize*
Output: A splitting rule *Rule* for the data

Pick any data point $\mathbf{t} \in \mathbf{V}$;
Let $\mathbf{x}$ be the farthest point from t in $\mathbf{V}$;
Let $\mathbf{y}$ be the farthest point from x in $\mathbf{V}$;
for $i = 1, 2, \ldots n$ **do**
    Project $\mathbf{v}_i$ onto the line spanned by $\mathbf{x}$ and $\mathbf{y}$;
    Let $fm(\mathbf{v}_i)$ be $\mathbf{v}_i$'s 1D coordinate value;
Sort the values $fm(\mathbf{v}_i)$ generating the list $s_1 \leq s_2 \leq \ldots \leq s_n$;
for $i = 1, 2, \ldots n$ **do**
    $\mu_1 = \frac{1}{i} \sum_{j=1}^{i} s_j$;
    $\mu_2 = \frac{1}{n-i} \sum_{j=i+1}^{n} s_j$;
    $c_i = \sum_{j=1}^{i} (s_j - \mu_1)^2 + \sum_{j=i+1}^{n} (s_j - \mu_2)^2$;
Find i that minimizes c_i and set $\theta = (s_i + s_{i+1})/2$;
$Rule(\mathbf{v}) := fm(\mathbf{v}) \leq \theta$;
return (*Rule*)

ture the important structure in the data at all. Reconsider standard CH-NMF. It corresponds to a tree of depth zero. In other words, the tree size is a tuning parameter governing HCH-NMF's complexity and one of its key advantages: *the diversified meaningfulness of its factorization*. The preferred strategy, as for example explained in [Hastie *et al.*, 2001], proceeds as follows. We grow a large tree T_0, stop the splitting process only when some minimum leaf size, say 200, is reached. Then, this tree is pruned in a post-processing step. We omit a detailed algorithmic description but rather briefly describe it now. The goal is to compute $k \geq l$ many basis vectors that reconstruct the data the best. We achieve this by successively merging neighboring leafs until we get k basis vectors. In each step, we find the two neighboring leafs that — if merged — produce the lowest reconstruction error over the covered examples. As this can be time consuming, we efficiently approximate it by always selecting the two neighboring leaves with highest resulting cluster accuracy when merging them.

In conclusion, HCH-NMF with a tree of depth 0 essentially coincides with CH-NMF. Moreover, the well known fact that if Z is any convex set that contains a convex hull $\mathbf{conv}(U)$ of a set U, in particular $Z = \mathbf{conv}(U \cup V)$, then $\mathbf{conv}(U) \subseteq Z$, see e.g. [Boyd and Vandenberghe, 2004], essentially proves the following correctness theorem.

Theorem 1 *HCH-NMF solves the convex NMF problem. It produces convex combinations of the data points in terms of basis vectors that minimize* (3) *but reside on convex hulls of clusters of data points.*

Due to the hierarchical decomposition of the data, HCH-NMF can adapt to the structure underlying the data as illustrated in Figs. 2 **(b,d,e)**. Indeed, it is akin to k-means. Because k-means clustering is a NP-hard optimization problem, see e.g. [Dasgupta and Freund, 2009b], this suggests that it is very unlikely that there exists an efficient algorithm for it.

4 Experiments

Our intention here is to investigate whether HCH-NMF can indeed find meaningful basis vectors in massive datasets and how it compares to CH-NMF. To this aim, we implemented both in Python using FastMap and the h5py HDF5 interface to deal with massive data. For optimization we used the cvxopt library by Dahl and Vandenberghe[2]. (H)CH-NMF offers many opportunities for parallelization.

[2] http://abel.ee.ucla.edu/cvxopt/

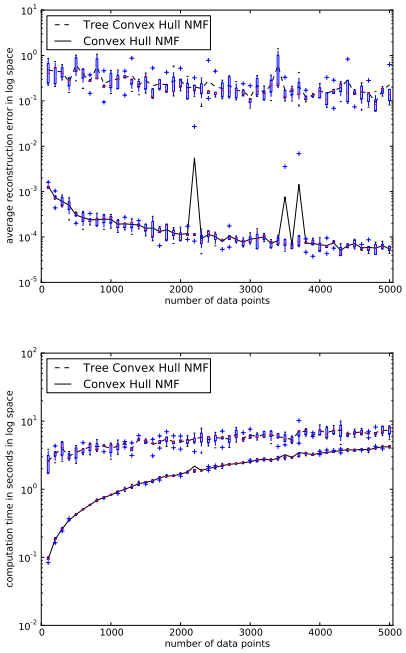


Figure 3: Boxplots of reconstruction errors **(top)** and computation times [sec.] **(bottom)** (both in log-space) of HCH-NMF and CH-NMF for varying numbers of synthetically generated data averaged over five reruns. As mentioned before, the Frobenius norm for CH-NMF has to be lower as CH-NMF approximates the convex hull of the complete data distribution and ignores the intrisic structure. Thus, for the purpose of interpretation and diversity this is probably not the right metric. (Best viewed in color.)

In the experiments, we only distributed the final reconstructions equally among all available cores. All experiments were ran on a standard Intel 3GHz computer with two cores. We report running times only for comparison of CH-NMF and HCH-NMF. Clearly, a C/C++ implementation would run several orders of magnitude faster.

We conducted four different experiments. To compare running time and reconstruction performance in a controlled setup, we compared (H)CH-NMF on synthetically generated data. Our main focus, however, are three additional experiments on massive real-world datasets, namely, publication histories of $760,000$ DBLP authors, 1.4 million activity profiles of guilds, and 4 million images of the Tiny image data-set [Torralba *et al.*, 2008]. For the sake of a better visualization, we show small trees.

Synthetic Data: Along the lines of [Ding *et al.*, 2009; Thurau *et al.*, 2009], we evaluated the mean reconstruction error and run-time performance using a varying number of data points sampled from three randomly positioned Gaussians in 2D. As already shown in [Thurau *et al.*, 2009], CH-NMF outperforms C-NMF for larger numbers of samples: it is several orders of magnitude faster while achieving competitive reconstruction errors. Therefore, we only compared HCH-NMF against CH-NMF. We varied the number of sampled data points ranging from 100 to 5000 in steps of 100. The maximum number of iterations for any numerical optimization was 100. The number of basis vectors was set to 12, i.e., we searched for overcomplete representations. The results averaged over 5 reruns are summarized in Fig. 3. As one can see, HCH-NMF produces an overhead

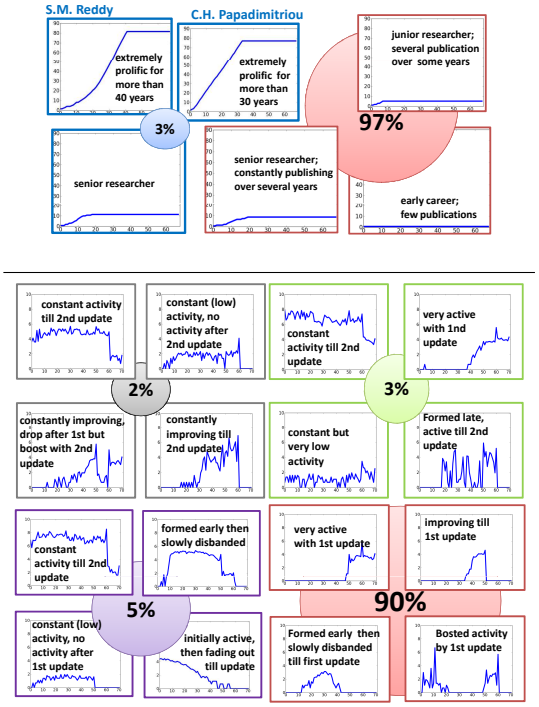


Figure 4: **(Top)** Clusters and corresponding basis vectors (histograms) found by HCH-NMF on the DBLP dataset. By describing the basis vectors, we gain an intuitively understandable description of academic careers: 97% of all authors are best described by the typical phases of an academic career. Indeed, there are renown, extremely prolific exceptions. Because the basis vectors are actual data points, we can identify them as Papadimitriou and Reddy. **(Bottom)** Clusters and corresponding basis vectors found by HCH-NMF on the World of Warcraft® dataset. (Best viewed in color.)

in running time for smaller sample sizes. For larger sample sizes, HCH-NMFs catches up. The reconstruction error is slightly higher than for CH-NMF but still lower than for k-means (as reported in [Thurau *et al.*, 2009]). The higher reconstruction error is due to the natur of convex hulls: any point within the convex hull can perfectly reconstructed. Hence, CH-NMF is a lower bound of HCH-NMF in terms of reconstruction error.

Bibliographic Analysis based on DBLP: Bibliographic databases such as DBLP[3] are a rich source of information. Here, we are interested in the question whether there are common patterns in the development of academic careers. To this aim, we extracted from DBLP the cumulative publication histograms of 757,368 authors, cf. Fig. 4**(Top)**. A publication histogram consists of the number of publications listed in DBLP in her first year, second year, and so on. We have cumulated the publications numbers of the years. The longest histogram we found spanned 68 years. To get equal length curves we filled missing years with 0. Following [Aitchison, 1982], we use logarithmic histogram values in our analysis. The idea is that the publication histogram of an author is a good descriptor for her activity and also to some extend for her success (but of course has not to imply impact/quality). A senior researcher, for example, is likely to have contributed over several years but there are exceptionally prolific authors. PhD students, on the other hand, may not have published many papers. We expected HCH-NMF to discover these variations. This was indeed the case as shown in Fig. 4 **(Top)**. The patterns found can be summarized as "*the majority of authors fall into one of the phases of a regular academic career (student, junior, senior) but of course there are illustrious exceptions.* It took 1 hour to compute the model, i.e., growing the tree and pruning it. CH-NMF took 45 minutes essentially yielding the union of all shown basis vectors, hence, giving the impression "*there is a Papadimitriou in all of us*".

Social Network World of Warcraft®: This dataset consists of recordings of the online appearance of characters in the computer game World of Warcraft®. It is assumed that World of Warcraft® has about 12 million p(l)aying customers. The game takes place in a virtual medieval fantasy environment. World of Warcraft® is often considered one large social platform which is used for chatting, team-play, and gathering. Compared to well known virtual worlds that mainly serve as chat platforms such as Second-life ®[4], World of Warcraft® is probably the *real* second life as it has a larger and more active (paying) user base. Moreover, a whole industry is developing around World of Warcraft®. It is estimated that 400.000 people world-wide are employed as gold-farmers, i.e. collecting virtual goods for online games and selling them over the Internet.

Players organize in groups, which are called guilds. Unlike groups known from other social platforms, such as Flickr, membership in a guild is exclusive. Obviously, the selection of a guild influences with whom players frequently interact. It also influences how successful players are in terms of game achievements. We assume that the level distribution among a guild is a good descriptor for its success and activity. For example, a guild of very experienced level 80 characters has a higher chance for achievements than a guild of level 10 players. Also, a level histogram gives an indicator for player activity over time. If players are continuously staying with a particular guild, we expect an equally distributed level histogram, as the characters are continuously increasing their level over time. The data was crawled from the publicly accessible site `www.warcraftrealms.com`. We viewed each character online appearance as a vote. Characters observations span a period of 4 years. Every time a character is seen online, he votes for the guild he is a member of according to his level. We accumulate the votes into a level-guild histogram, going from level 10 (level 1-9 are excluded) to level 80 (the highest possible level). Players advance in level by engaging in the game, i.e. completing quests or other heroic deeds. Following [Aitchison, 1982], we use logarithmic histogram values in our analysis. In total, we collected 150 million votes of 18 million characters belonging to 1.4 million guilds.

Running HCH-NMF took about 2.5 hours and revealed some very interesting patterns as shown in Fig. 4 **(Bottom)**. As for CH-NMF, see [Thurau *et al.*, 2009], we can also spot singular events, in our case large *updates* to the game content (this is a regular procedure that makes novel content available and also allows a further advancement in character level). Apparently, large updates to the game can result in a restructuring of social groups. More interestingly, HCH-NMF allows us to descriptions of clusters. For instance, 90% of the data can be described in terms of just

[3] `http://www.informatik.uni-trier.de/~ley/db/`

[4] `http://secondlife.com/`

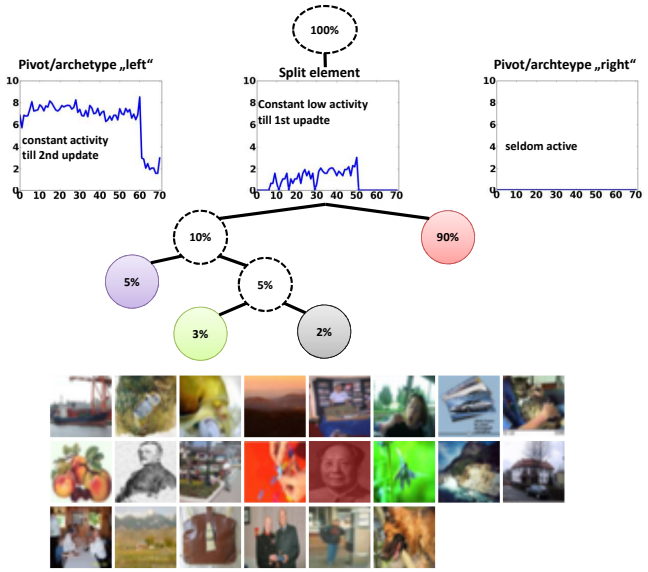


Figure 5: **(Top)** HCH-NMF tree as well as the pivot and split elements for the first split on the World of Warcraft® dataset. The "90%" cluster in Fig. 4 captures the data "between" the split element and the "right" pivot element. Thus, the majority of guilds is actually close to the "low constant activity" or "seldom activity" guilds. Both patterns are well captured by combining the corresponding four basis vectors in Fig. 4. **(Bottom)** HCH-NMF's split elements for the 4 million tiny images are natural images. (Best viewed in color.)

four basis vectors: "*formed early then slowly disbanded till 1st update*", "*improving till 1st update*", "*very active with 1st update*", and "*Boosted activity with 1st update*". That was surprising. We knew that most guilds are rather *seldom active*, see [Thurau *et al.*, 2009]. Therefore, we "zoomed" into the model, a feature of HCH-NMF not supported by CH-NMF. Specifically, we had a look at the pivot and split elements of the first level of the induced tree, see Fig. 5 **(Top)**. This revealed that most of the 90% are actually "*seldom active*": they lay between the "constantly active till 2nd update" and "seldom active" guild on the FastMap line. The four basis vectors of the 90% cluster together are archetypical guilds to reconstruct this pattern. Running CH-NMF took about 2 hours and produced essentially the same basis vectors. The major difference was that it used the "*seldom active*" guild directly and did not factorize it.

Massive Image Collection: Our final experiment applies HCH-NMF to a subset of 4 million images of Torralba *et al.*'s 80 million tiny images [Torralba *et al.*, 2008]. The images are represented as 384 dimensional GIST feature vector. The result of running HCH-NMF is shown in Fig. 6. As already reported in [Thurau *et al.*, 2009], some of the basis vectors discovered show a geometric similarity to Walsh filters that are found among the principal components of natural images [Heidemann, 2006]. This suggests that the extremal points in this large collection of natural images are located close to the principal axes of the data. On the other hand, the split elements as shown in Fig. 5 **(Bottom,)** are mostly "realistic" images. This suggests that they are located in the center of the data/clusters. In contrast to CH-NMF, HCH-NMF grouped the basis vectors into meaningful clusters. From left to right, the basis vectors, i.e., the columns, capture different aspects of the images such as dark-light-dark, horizontal-cross/circle-vertical, complex-plain-complex. Overall, running HCH-NMF took only 23 hours and 37 minutes. This is remarkable as we used an external USB hard disk and the running time can be considerably reduced using an internal hard disk and implementing HCH-NMF fully in C/C++.

5 Conclusion

We have introduced hierarchical convex-hull NMF. It seeks to leverage convex NMF by automatically adapting to the geometric structure of the data. It is fast and straightforward to implement, provably solves the convex NMF problem, combines the interpretability of both convex NMF as well as hierarchical decomposition methods, and scales well to massive, high-dimensional datasets. These contributions advance the understanding of descriptive analytics of massive, high-dimensional datasets and is an encouraging sign that applying NMF techniques in the wild, i.e., on hundreds of millions of data points may not be insurmountable.

There are several interesting avenues for future work. One is the application of HCH-NMF to other challenging datasets, such as Wikipedia, Netflix, Facebook, or the blogsphere, and to use it for applications such as collaborative filtering. For the latter case, it is interesting to develop bottom up HCH-NMF variants and to investigate the missing values case. Another important avenue is parallelization. HCH-NMF suggests a natural data-driven parallelization: the training set is partitioned into subsets associated with separate processors. Finally, HCH-NMF is highly relevant for high-dimensional classification problems. Here, it is infeasible to include enough training samples to cover the class regions densely. As Cevikalp *et al.* [2008] have recently pointed out, irregularities in the resulting sparse sample distributions cause local classifiers such as nearest neighbors and kernel methods to have irregular decision boundaries. One solution is to "fill in the holes" by building a convex model of the regions spanned by the training samples. Using HCH-NMF, we even take the geometric structure of each class into account.

Acknowledgment The authors would like to thank A. Torralba, R. Fergus, and W. T. Freeman for making the tiny images freely available and S. Zayakh for preparing the DBLP dataset. M. Wahabzada and K. Kersting were supported by the Fraunhofer ATTRACT Fellowship STREAM.

References

[Aitchison, 1982] J. Aitchison. The Statistical Analysis of Compositional Data. *J. of the Royal Statistical Society B*, 44(2):139–177, 1982.

[Boyd and Vandenberghe, 2004] S. Boyd and L. Vandenberghe. *Convex Optimization*. Camb. Univ. Press, 2004.

[Cai *et al.*, 2008] Deng Cai, Xiaofei He, Xiaoyun Wu, and Jiawei Han. Non-negative matrix factorization on manifold. In *International Conference on Data Mining*, pages 63–72. IEEE, 2008.

[Cevikalp *et al.*, 2008] H. Cevikalp, B. Triggs, and R. Polikar. Nearest hyperdisk methods for high-dimensional classification. In *Proceedings of the 25th International Conference on Machine Learning (ICML-08)*, 2008.

Figure 6: HCH-NMF clusters (columns) and corresponding basis vectors (images in the columns) of the 4 million tiny images. From left to right, the basis vectors capture different aspects of the images such as dark-light-dark, horizontal-cross/circle-vertical, complex-plain-complex. (Best viewed in color.)

[Cutler and Breiman, 1994] A. Cutler and L. Breiman. Archetypal Analysis. *Technometrics*, 36(4):338–347, 1994.

[Dasgupta and Freund, 2009a] S. Dasgupta and Y. Freund. Random projection trees and low dimensional manifolds. In R.E. Ladner and C. Dwork, editors, *Proceedings of the 40th Annual ACM Symposium on Theory of Computing (STOC-08)*, pages 537–546, 2009.

[Dasgupta and Freund, 2009b] S. Dasgupta and Y. Freund. Random projection trees for vector quantization. *IEEE Transactions on Information Theory*, 55:3229–3242, 2009.

[de Berg *et al.*, 2000] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry*. Springer, 2000.

[Ding *et al.*, 2009] C.H.Q. Ding, T. Li, and M.I. Jordan. Convex and Semi-Nonnegative Matrix Factorizations. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2009. Accepted for publication.

[Donoho and Tanner, 2005] D.L. Donoho and J. Tanner. Neighborliness of Randomly-Projected Simplices in High Dimensions. *Proc. of the Nat. Academy of Sciences*, 102(27):9452–9457, 2005.

[Faloutsos and Lin, 1995] C. Faloutsos and K.-I. Lin. FastMap: A Fast Algorithm for Indexing, Data-mining and Visualization of Traditional and Multimedia Datasets. In *Proc. ACM SIGMOD*, 1995.

[Golub and van Loan, 1996] G.H. Golub and J.F. van Loan. *Matrix Computations*. Johns Hopkins University Press, 3rd edition, 1996.

[Halevy *et al.*, 2009] A.Y. Halevy, P. Norvig, and F. Pereira. The unreasonable effectiveness of data. *IEEE Intelligent Systems*, 24:8–12, 2009.

[Hall *et al.*, 2005] P. Hall, J. Marron, and A. Neeman. Geometric Representation of High Dimension Low Sample Size Data. *J. of the Royal Statistical Society B*, 67(3):427–444, 2005.

[Hastie *et al.*, 2001] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2001.

[Heidemann, 2006] G. Heidemann. The principal components of natural images revisited. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(5), 2006.

[Hueter, 1999] I. Hueter. Limit Theorems for the Convex Hull of Random Points in Higher Dimensions. *Trans. of the American Mathematical Society*, 351(11):4337–4363, 1999.

[Jolliffe, 1986] I.T. Jolliffe. *Principal Component Analysis*. Springer, 1986.

[Kim and Park, 2008] Jingu Kim and Haesun Park. Toward faster nonnegative matrix factorization: A new algorithm and comparisons. In *International Conference on Data Mining*, pages 353–362. IEEE, 2008.

[Kolda and Sun, 2008] Tamara G. Kolda and Jimeng Sun. Scalable tensor decompositions for multi-aspect data mining. In *International Conference on Data Mining*, pages 363–372. IEEE, 2008.

[Lee and Seung, 1999] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–799, 1999.

[Mairal *et al.*, 2010] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online learning for matrix factorization and sparse coding. *JMLR*, 11:19–60, 2010.

[Ostrouchov and Samatova, 2005] G. Ostrouchov and N.F. Samatova. On FastMap and the convex hull of multivariate data: toward fast and robust dimension reduction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1340–1434, 2005.

[Suvrit, 2008] Sra Suvrit. Block-iterative algorithms for non-negative matrix approximation. In *ICDM*, pages 1037–1042. IEEE, 2008.

[Talwalkar *et al.*, 2008] Ameet Talwalkar, Sanjiv Kumar, and Henry Rowley. Large-scale manifold learning. In *Computer Vision and Pattern Recognition*. IEEE, 2008.

[Thurau *et al.*, 2009] C. Thurau, K. Kersting, and C. Bauckhage. Convex non-negative matrix factorization in the wild. In H. Kargupta and W. Wang, editors, *Proceedings of the 9th IEEE International Conference on Data Mining (ICDM-09)*, 2009.

[Torralba *et al.*, 2008] Antonio Torralba, Rob Fergus, and William T. Freeman. 80 Million Tiny Images: A Large Data Set for Nonparametric Object and Scene Recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 30(11):1958–1970, 2008.

[Ziegler, 1995] G.M. Ziegler. *Lectures on Polytopes*. Springer, 1995.

Quantitatives Frequent-Pattern Mining über Datenströmen

Daniel Klan, Thomas Rohe
Department of Computer Science & Automation
TU Ilmenau, Germany
{*first.last*}@tu-ilmenau.de

Abstract

Das Aufdecken unbekannter Zusammenhänge zählt zu einer der wichtigsten Aufgaben im Data Mining. Für das Problem des Frequent Pattern Mining über statischen Daten finden sich daher in der Literatur eine Vielzahl an Lösungen. Die Integration von Sensorik in nahezu jeden Lebensbereich führt allerdings zu Datenmengen, welche mittels der klassischen Verfahren zumeist nicht mehr bewältigt werden können. Ein Paradigmenwechsel hin zur Datenstrom-Verarbeitung ist oftmals unumgänglich. Ein interessantes Problem, welches im Zusammenhang mit der Verarbeitung von Sensordaten auftritt ist der prinzipiell stetige Wertebereich von Messungen. Die bekannten Lösungen sind für die Analyse von kontinuierlichen Daten über stetigen Wertebereichen nur bedingt geeignet. Im folgenden soll mit dem FP^2-Stream ein entsprechendes Verfahren für die Analyse quantitativer häufiger Muster über Datenströmen präsentiert werden.

1 Einleitung

Das Finden häufiger Muster (*frequent pattern mining*) ist Grundlage für eine Vielzahl von Data Mining Problemen (zum Beispiel der Korrelationsanalyse, oder dem Finden von Sequenzen oder Perioden innerhalb des Datenstromes). Die populärste Anwendung für das Frequent Pattern Mining ist das Assoziation Rule Mining, bei welcher aus Transaktionen Werte extrahiert und in Relation zueinander gesetzt werden. Eine typische Anwendung ist die Analyse von Supermarkt-Transaktionen, bei welcher das Kaufverhalten der Kunden analysiert wird. Ziel ist dabei die Detektion von Regeln, welche das Kaufverhalten einer repräsentativen Menge von Kunden wiederspiegeln und anhand derer anschließend die Verkaufsprozesse optimiert werden können. Eine mögliche Regel ist "Bier $\rightarrow$ Chips (10%)", welche besagt, dass 10 Prozent aller Kunden, die Bier gekauft haben auch Chips kaufen.

Wie die meisten Data Mining Verfahren, so wurde auch das Frequent Pattern Mining ursprünglich für statische Datenbestände entwickelt. In den letzten Jahren wurden diese Verfahren an das Datenstrom-Paradigma angepasst [5; 8]. In Interessantes Beispiel für die Anwendung der Assoziation Rule Mining über Datenströmen ist in [4] beschrieben. Die Autoren ersetzen dabei fehlende Werte in einem WSN anhand zuvor abgeleiteter Assoziationsregeln.

In einer Vielzahl von Anwendungen liegen die Daten zur Analyse nicht in Form kategorischer, sondern als quantitative Attribute vor. Die Anwendung der bekannten Frequent Pattern Mining Verfahren führt hierbei oftmals nicht zu den gewünschten Zielen bzw. der Berechnungsaufwand ist nicht vertretbar. Basierend auf dieser Feststellung entwickelten die Autoren in [11] erstmals ein Verfahren für das Finden von Regeln, welche sowohl mit kategorischen Attributen als auch mit quantitativen Attributen umgehen kann. Das vorgestellte Verfahren bildet dabei quantitative auf kategorische Attribute ab. Die quantitativen Attribute werden im wesentlichen durch Intervalle dargestellt, welche eine Menge von numerischen Werten aufnehmen können. Auf Basis dieser Definition präsentierten die Autoren einen angepassten Apriori-Algorithmus, welcher den Suchraum zerlegt (diskretisiert) und anschließend durch Kombination der Teil-Intervalle Regeln findet.

Es existieren eine Vielzahl von Anwendungsszenarien in denen die Extraktion quantitativer Regeln bzw. das Finden quantitativer Frequent Itemsets über Datenströmen von Interesse sind (z.B. Gebäudeüberwachung). Die in der Literatur präsentierten Verfahren beziehen allerdings sich ausschließlich auf das Finden von Regeln über statischen Datenbeständen. Für ein Verarbeitung von Datenströmen sind diese ungeeignet. Zudem weisen die meisten Verfahren Beschränkungen auf, derart dass quantitative Attribute lediglich im Regel-Kopf auftreten dürfen.

Die Arbeit ist im weiteren wie folgt aufgeteilt. Zunächst folgt in Abschnitt 2 ein kurzer Überblick über existierende Arbeiten zum Thema. Anschließend wird in Abschnitt 3 auf für das Verständnis notwendige Grundlagen eingegangen. In Abschnitt 4 folgt eine ausführliche Beschreibung des entwickelten Verfahrens, welche in Abschnitt 5 evaluiert werden soll.

2 Verwandte Arbeiten

Eines der ersten Verfahren zum Finden von häufigen Mustern war das von Aggrawal et. al präsentierte *Apriori*-Verfahren [1]. Die Grundlage des Ansatzes bildet die Annahme, dass sich häufige Muster ausschließlich aus häufigen Teilmustern zusammensetzen können. Basierend auf dieser Heuristik erzeugt das Verfahren alle möglichen $k-itemset$ Kandidaten aus der Menge aller $k-1$-itemset Kandidaten. Aufgrund der wiederholten Analyse der Daten handelt es sich hierbei um ein *multi-pass* Verfahren, welches seine Anwendung insbesondere in der Verarbeitung von endlichen Datenmengen findet.

Auf Basis des Apriori-Algorithmus wurden eine Vielzahl an weiteren Algorithmen entwickelt, welche unter anderem die Performance-Steigerung oder die Adaption an Datenströme zum Ziel hatten. So stellen die Autoren in [3] ein effizientes Verfahren zum Finden von optimalen links-

seitigen quantitativen Regeln vor. Der Schwerpunkt ist dabei die Entwicklung eines Verfahrens zur Berechnung der optimalen Bereiche in linearer Zeit unter der Annahme das die Daten sortiert sind. Für denn Fall, dass eine Sortierung der Daten nicht möglich ist, wurde zusätzlich ein randomisiertes Verfahren präsentiert, welches die Daten in Buckets teilt und anschließend auf diesen die Suche durchführt. Die Autoren in [9] präsentierten mit dem DHP (*Direct Hashing and Pruning*) ein Verfahren, dessen Ziel die Optimierung der Anzahl an Kandidaten ist. Das Verfahren setzt dabei auf eine Hash-Tabelle für die Identifikation von Kandidaten mit einem Support größer dem geforderten.

Han et. al präsentieren in [5] mit dem FP-Tree (*frequent pattern tree*) einen Prefix-Baum zur effizienten Speicherung häufiger Muster. Zusätzlich zu dieser Datenstruktur stellen sie mit dem FP-Growth Algorithmus ein Verfahren zum Finden von häufigen Mustern auf Basis des FP-Tree vor. Im Gegensatz zum Apriori Verfahren verzichtet das FP-Growth Verfahren vollständig auf die Generierung von Kandidaten. Auch sind lediglich zwei Durchläufe über die zu prüfende Datenmenge ausreichend.

FP-Mining über Datenströmen stellt eine besondere Herausforderung dar. Neben dem in [5] beschriebenen FP^2-Stream existieren auch eine Vielzahl weiterer Lösungen. Der in [12] vorgestellte Compact Pattern Tree (CPT) zum Beispiel verwendet gleitende Fenster (analog dem FP^2-Stream und dem DSTree [7]). Die Effizienz des Verfahrens wird dabei durch eine zusätzliche Restrukturierungs Phase erreicht, während der der CPT in Abhängigkeit der Itemset-Häufigkeiten mit dem Ziel einer hohen Kompaktheit umsortiert wird. In [8] beschreiben die Autoren neben dem Sticky-Sampling und Lossy-Counting mit BTS(Buffer-Trie-SetGen) ein Verfahren, welches die Häufigkeiten von Itemsets mittels einer Gitter-Strukur speichert.

Die erste Arbeit die sich mit dem Problem der quantitativen Muster Erkennung beschäftigt stammt von Aggrawal [11]. Das vorgestellte Verfahren basierte dabei im wesentlichen auf einem Apriori-Verfahren, welches die partitionierten Wertebereiche der einzelnen Attribute in geeigneter Weise kombiniert.

Die Autoren in [2] betrachten quantitativ/kategorische Assoziationsregeln. Eine Regel wird signifikant angesehen, wenn sich deren Mittelwert in Relation zur Menge aller Transaktionen ohne dieses Regel als signifikant herausstellt. Zur Berechnung des Signifikanzlevels ziehen sie dabei den Steiger Z-Test heran. Als Null-Hypothese nehmen sie an, das die Mittelwerte beider Teilmengen gleich sind. Wird diese Null-Hypothes abgelehnt (mit einer Konfidenz von 95%), so wird die gefundene Regel als signifikant unterschiedlich von der Restmenge der Transaktionen angenommen.

3 Vorbetrachtung

Zunächst müssen verschiedene Begrifflichkeiten eingeführt werden, welche für das Verständnis des im weiteren präsentierte Verfahren notwendige sind.

Im folgenden bezeichnet a_i ein Attribut. $a_i\langle v\rangle$ entspricht dem zum Attribut a_i gehörender Wert v. Das Tripel $(a_i, l, r) = a_i(l, r)$ bezeichnet das *quantitative Attribut* a_i, welches Werte $v_1, v_2, ...$ im Intervall (l, r) aufnimmt (l bezeichnete die linke und r die rechte Intervallgrenze). Ein *kategorisches Attribut*, d.h. ein einelementiges Attribut, kann durch ein quantitatives Attribut repräsentiert werden, dessen linke und rechte Intervallgrenze gleich sind, d.h. $a_i(l, l)$ bezeichnet das kategorische Attribut, welches nur den Wert

	a_1	a_2		a_1	a_2
t_1	22	100	t_6	21.5	0
t_2	22.5	-	t_7	21	-
t_3	-	100	t_8	-	100
t_4	22.5	-	t_9	22	-
t_5	21.5	0	t_{10}	22.5	100

Abbildung 1: Beispiel Transaktionen

itemset	$freq$	$supp$
$\{\langle a_1(20, 21.5]\rangle\}$	3	0.3
$\{\langle a_1(20, 23]\rangle\}$	8	0.8
$\{\langle a_2(0, 50]\rangle\}$	4	0.4
$\{\langle a_2(50, 100]\rangle\}$	4	0.4
$\{\langle a_1(20, 21.5]\rangle, \langle a_2(0, 50]\rangle\}$	2	0.2
$\{\langle a_1(20, 23]\rangle, \langle a_2(0, 50]\rangle\}$	2	0.2
$\{\langle a_1(20, 23]\rangle, \langle a_2(50, 100]\rangle\}$	2	0.2

Abbildung 2: Beispiel Itemsets

l repräsentiert. Das Attribut $a_i(l, r)$ wird im folgenden als Item bezeichnet. $\mathcal{I}$ bezeichnet die Menge aller Items. Die Menge $X = \{a_1(l, r), ..., a_k(l, r)\} \subseteq \mathcal{I}$, with $a_i \neq a_j$ bezeichnet ein Itemset (oder auch k-Itemset).

D bezeichnet eine Menge von Transaktionen. Eine Transaktion $d_{\Delta t} \in D$ ist eine Menge von Attributwerten $a_i\langle v\rangle$ im Zeitintervall Δt. Jede Transaktion $d_{\Delta t} = \{a_i\langle v\rangle, ..., a_j\langle v\rangle\}$ kann auf ein Itemset $\mathcal{I}$ abgebildet werden, d.h. für jeden Wert $a_i\langle v\rangle$ existiert ein Item $a_i(l, r)$ in $\mathcal{I}$ mit $a_i\langle v\rangle \in a_i(l, r)$.

Die Häufigkeit $freq(X, D)$ eines Itemsets X bezeichnet die Anzahl der Transaktionen D im Zeitintervall Δt, die auf das Itemset abgebildet werden können. Der $supp(X, D)$ eines Itemsets ist definiert als der prozentuale Anteil an Transaktionen, welcher auf das Itemset entfällt. Üblicherweise sind nur häufige Itemsets von Interesse. Ein Itemset wird als häufig bezeichnet, wenn dessen Häufigkeit einen vordefinierten Schwellwert $minsupp$ überschreitet.

Signifikanz und Informationsdichte Ziel des quantitativen Frequent Itemset Mining ist das Finden von zusammenhängenden Items, deren Informationsgehalt sich *signifikant* von dem aller anderen Items unterscheidet. Ein quantitatives Item ist genau dann signifikant, wenn dessen Informationsdichte größer ist, als die Informationsdichte der alternativen Items. Die Informationsdichte eines Items $a_i(l, r)$ ist dabei wie folgt definiert

$$density(a_i(l, r)) = \frac{freq(a_i(l, r))}{dist(l, r)} \tag{1}$$

wobei $dist(l, r) = abs(l - r)$ die Distanz zwischen den beiden Intervallgrenzen des Items bezeichnet. Entsprechend dieser Definition wird ein Itemset $\mathcal{I}$ genau dann als signifikant bezeichnet, wenn alle Items dieses Itemsets signifikant sind.

Generalisierung Ein Itemset $\hat{X}$ wird genau dann als *Generalisierung* von X bezeichnet, wenn $\hat{X}$ aus den selben Items wie X besteht und es gilt

$$\forall a_i(l, r) \in X : a_i(l, r) \in X \wedge a_i(l', r') \in \hat{X} \Rightarrow l' \leq l \leq r \leq r'$$

D.h. alle Transaktion, welche sich auf X abbilden lassen, können ebenso auf $\hat{X}$ abgebildet werden. Im folgenden soll ein kurzes Beispiel die eben beschriebenen Zusammenhänge verdeutlichen.

Beispiel 1 *Es wird ein Datenstrom angenommen bestehend aus zwei Attributen a_1 (Temperatur) und a_2 (Bewegung) angenommen. Im Zeitintervall Δt wurden die in Tabelle 3 dargestellten 10 Transaktionen festgestellt. Weiterhin werden die folgenden Items angenommen:*

$$a_1(20, 21.5], a_1(20, 23], a_2(0, 50] \text{ und } a_2(50, 100]$$

Basierend auf diesen (quantitativen) Items lassen sich über den in Tabelle 3 abgebildeten Transaktionen die in Tabelle 3 dargestellten (quantitativen) Itemsets ermitteln. Die Tabelle zeigt die entsprechenden Häufigkeiten und den Support, den die einzelnen Itemsets aufweisen. Es lassen sich die folgenden Dichten für die Items ermitteln: $density(a_1(20, 21.5])) = 2$, $density(a_1(21.5, 23]) = 3.3$ und $density(a_1(20, 23]) = 2.7$. Bei dem einelementigen Itemset $\{\langle a_1(20, 23]\rangle\}$ handelt es sich um eine Generalisierung der Itemsets $\{\langle a_1(20, 21.5]\rangle\}$ und $\{\langle a_1(21.5, 23]\rangle\}$.

4 FP^2-Stream

Die meisten in der Literatur zu findenden Verfahren [11; 2], welche quantitative Attribute betrachten, zerlegen den Wertebereich für ein Attribut in äquidistante Intervalle, welche anschließend derart miteinander kombiniert werden, dass sie den geforderten Kriterien (minimaler Support und Signifikanz) genügen. Zu den wesentlichen Problemen dieser *bottom-up* Strategie zählen dabei das Finden einer geeignete Zerlegung [11] bzw. das möglichst effiziente generieren zusammenhängender Items.

Die Zerlegung gefolgt von einer kostenintensiven Item-Rekonstruktion ist auf Datenströme nicht bzw. nur beschränkt anwendbar. Alle Verfahren, welche das Prinzip der Kombination von Teilintervallen einsetzen, basieren dabei auf dem Apriori-Algorithmus, welcher mehrere Durchläufe benötigt und daher für die Analyse über Datenströmen nur bedingt geeignet ist.

Im folgenden soll der FP^2-Stream vorgestellt werden, ein Speicher-effizientes Verfahren, welches für die Analyse über Datenströmen geeignet ist. Die Itemsets werden beim FP^2-Stream in einem Prefix-Baum (ähnlich dem FP-Tree von Han et al. [5]) verwaltet. Das Ziel des vorgestellten Verfahrens ist anschließend die kontinuierliche Verfeinerung der Items (*top-down*-Strategie) und den daraus aufgebauten Itemsets mit dem Eintreffen neuer Transaktionen, so dass diese den geforderten Kriterien genügen.

Im weiteren soll zunächst die Grundlegende Datenstruktur beschrieben werden, anschließend folgt eine Beschreibung der Algorithmen zum Einfügen von Transaktionen und zum Optimieren der Datenstruktur.

4.1 Datenstruktur

Der FP^2-Stream verwaltet die häufigsten Muster in einem Prefix-Baum. Eine Header-Tabelle enthält alle Itemsets, welche sich gegenwärtig im Prefix-Baum befinden. Der Pfad von der Wurzel bis zu einem Knoten im Prefix-Baum repräsentiert ein Itemset. Zusätzlich ist in jedem Knoten des Baumes ein Zeitfenster eingebettet, welches die Häufigkeiten in den letzten k Zeiträumen aufnimmt. Im FP^2-Stream werden hierzu gleitende Fenster eingesetzt.

Weiterhin sind alle Knoten, welche das gleiche Item repräsentieren untereinander über Listen miteinander verbunden. Das entsprechende Item der Header-Tabelle verweist dabei auf das erste Element dieser Liste. Jedes Item der Header-Tabelle enthält zusätzlich ein *Equi-Width Histogramm*, welches einen approximativen Überblick über die Häufigkeitsverteilung der zuletzt eingefügten Werte in die Items gibt. Abbildung 3 zeigt einen Beispiel-FP^2-Stream. Dem Beispiel liegen die in Tabelle 3 beschriebenen Transaktionen zugrunde. Das Attribut a_2 wurde bereits einer Verfeinerung unterzogen.

Input: Menge von Transaktionen D
Output: Menge von Häufigen Itemsets

1 Initialisiere FP^2-Tree als leer;
2 Stelle min und max für jedes Attribut a_i fest und lege die entsprechenden knoten im FP^2-Tree an;
3 Füge die Transaktionen von Batch b_0 in FP^2-Tree ein;
4 Übernehme Knotengrenzen und Häufigkeiten aus dem FP^2-Tree in den FP^2-Stream;
5 **while** *Batch* b_i, $i > 0$ **do**
6 Initialisiere FP^2-Tree mit den Knoten und Intervallgrenzen des FP^2-Stream;
7 Sortiere alle Transaktionen aus aktuellem Batch in FP^2-Tree ein (falls notwendig, füge neue Knoten hinzu bzw. erweitere existierende Knoten);
8 Übertrage alle Knoten aus dem FP^2-Tree in den FP^2-Stream;
9 Führe eventuell notwendige Split Operationen auf dem FP^2-Stream aus;
10 Führe eventuell notwendige Merge Operationen auf dem FP^2-Stream durch;
11 Lese alle häufigen Itemsets aus dem FP^2-Stream aus (FP^2-Growth);

Algorithm 1: FP^2-Stream Algorithmus

4.2 Einfügen neuer Transaktionen

Das Einfügen neuer Transaktionen in den FP^2-Stream erfolgt Batchweise (ein Batch b bezeichnte die Zusammenfassung einer Menge von $|b|$ Transaktionen). Neue Transaktionen werden beim Einfügen nicht direkt in den FP^2-Stream integriert, sondern zuvor in einen FP^2-Tree eingefügt. Der FP^2-Tree entspricht dabei im wesentlichen dem FP^2-Stream, wobei die Knoten jedoch nicht über Zeitfenster verfügen (der FP^2-Tree verwaltet lediglich die zu einem Batch gehörenden Musterhäufigkeiten). Die Implementierung des FP^2-Tree erfolgt in Form von "Schattenknoten", welche in die Knoten des FP^2-Stream integriert werden. Somit fallen für das Anlegen des FP^2-Tree keine zusätzliche Kosten an. Es muss lediglich beim ersten Einfügen einer Transaktion in einen neuen Batch der entsprechende "Schattenknoten" angelegt werden.

Analog dem FP-Stream wird das Einfügen des ersten Batches b_0 getrennt von der Behandlung aller weiteren Batches betrachtet. Zum Zeitpunkt des Einfügens von b_0 in den FP^2-Stream ist kein Wissen über die genaue Verteilung der Stromdaten vorhanden bzw. es stehen lediglich die Informationen aus dem ersten Batch zur Verfügung. Zunächst wird daher für jedes Attribute i ein Item $a_i(min, max)$ derart angelegt, dass min dem minimalen Wert in b_0 und max dem maximalen Wert in b_0 des Attributes a_i im ersten Batch entspricht. Die Items werden anschließend nach der Häufigkeit ihres Auftretens im ersten Batch sortiert und in den FP^2-Tree eingefügt. Häufige Items werden Wurzelnah eingefügt. Da im ersten Batch ein Attribut den kompletten Wertebereich eines Datenstromes überdeckt, kann es ausschließlich durch das vollständige Fehlen von Werten innerhalb von Transaktionen zu Unterschieden in den Häufigkeiten der einzelnen Attribute kommen.

Nachdem der erste Batch erfolgreich eingefügt wurde,

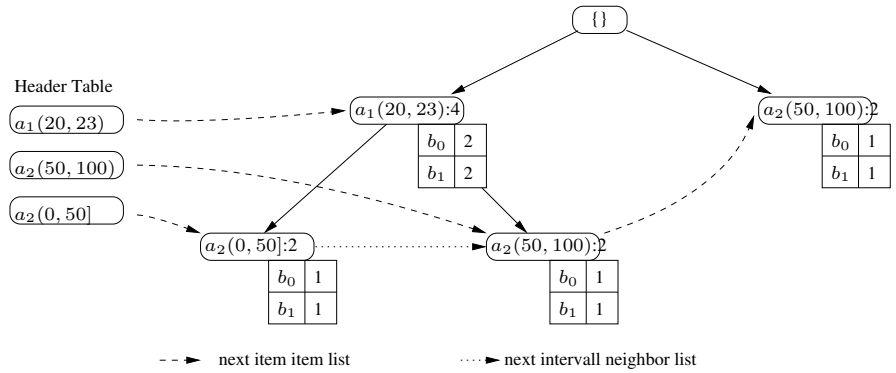


Abbildung 3: Beispiel FP^2-Stream

werden alle weiteren Batches gleich behandelt und folgt nachstehendem Schema:

- Es existiert bereits ein Knoten, welcher das Itemset repräsentiert. Die Frequenz des entsprechenden Knotens im FP^2-Tree wird um 1 inkrementiert.
- Es existiert kein Knoten, welcher das Itemset repräsentiert. Es muss ein neuer Knoten im FP^2-Tree angelegt werden:
  - Der neu anzulegende Knoten wird sowohl auf der linken, als auch auf der rechten Seite von existierenden Knoten eingeschlossen: Als Intervallgrenzen für den neuen Knoten werden die Grenzen der benachbarten Knoten gewählt. D.h. es wird ein Knoten $a_i(r, l')$ zwischen den beiden begrenzenden Knoten $a_i(l, r)$ und $a_i(l', r')$ angelegt.
  - Der einzufügende Wert über- bzw. unterschreitet alle bisher eingefügten Werte: Existiert ein Knoten im FP^2-Tree, welches noch keinem Knoten im FP^2-Stream entspricht und dessen Intervallgrenzen sich derart erweitern lassen, dass es den Wert aufnehmen kann, dann werden die Grenzen dieses Knotens angepasst und die Frequenz entsprechend inkrementiert. Andernfalls muss ein neuer Knoten mit dem neuen Wert als Minimum bzw. Maximum als Grenze angelegt werden.

Wurden ausreichend Transaktionen in den FP^2-Tree eingefügt, dann kann dieser in den FP^2-Stream integriert werden. Zunächst werden hierzu alle Knoten des FP^2-Tree entfernt, welche dem vorab definierten Schwellwert ϵ nicht genügen. Anschließend werden die Häufigkeiten der einzelnen Knoten in den FP^2-Stream übernommen. Hierzu werden in die Zeitfenster aller Knoten des FP^2-Stream neue Zeitslots eingeführt. Sollten im FP^2-Stream Knoten vorhanden sein, welche durch die Integration des FP^2-Tree keine Aktualisierung erfahren, so sind deren Häufigkeiten für diesen Zeitslot 0. Sind während der Verarbeitung eines Batches neue Knoten im FP^2-Tree hinzugekommen, so müssen diese ebenfalls in den FP^2-Stream übernommen werden.

Wurden die Transaktionen eines Batches erfolgreich in den FP^2-Stream eingefügt, dann wird anschließend geprüft, inwieweit sich Items verfeinern lassen um eine möglichst hohe Informationsdichte zu erhalten. Die Verfeinerung ist hierbei ein zweistufiger Prozess. In einem ersten Schritt werden die Items verfeinert, wenn deren Füllgrad zu hoch ist. Der zweite Schritt ist das Mischen von Items, wenn sich diese generalisieren lassen. Beide Prozesse sollen im weiteren beschrieben werden.

4.3 Item-Split

Als wesentliches Kriterium für schlecht approximierte Items wird die Dichteverteilung innerhalb eines Items herangezogen. Sind die in ein Item eingefügten Werte ungleichmäßig verteilt, dann wurden die Grenzen für dieses Item schlecht gewählt. Die Bestimmung der Ungleichverteilung erfolgt dabei über die Schiefe den in der Header Tabelle mitgeführten *Equi-Width-Histogrammen*. Die Schiefe eines Item $a_i(l, r)$ über einem Histogramm lässt sich wie folgt bestimmen:

$$s(a_i(l,r))_n = \frac{\max\{h(a_i(l,r),j)\}_{j=1}^{n} - \min\{h(a_i(l,r),j)\}_{j=1}^{n}}{\sum_{j=1}^{n} h(a_i(l,r),j)}$$

wobei n die Anzahl der Buckets in den Histogrammen bezeichnet. $h(a_i(l, r), j)$ bezeichnet die Häufigkeit im j-ten Bucket des Histogramms für das Item $a_i(l, r)$.

Überschreitet die Schiefe innerhalb des Items $a_i(l, r)$ einen vorab definierten Schwellwert $maxskew$, d.h. $s(a_i(l, r)) > maxskew$, dann wird das Item in zwei disjunkte Items gesplittet. O.B.d.A. werden im folgenden Items immer in Items mit links offenem Intervall zerlegt. Für den Split wird ein Median-basierter Ansatz verwendete, dessen Ziel die Berücksichtigung der realen Dichteverhältnisse innerhalb eines Items ist. Hierzu wird der Median über die einzelnen Buckets der Histogramme bestimmt. Der Split erfolgt anschließend auf Basis des Mittelpunktes des Median-Buckets, d.h. $a_i(l, r)$ wird in die beiden Items $a_i(l, median/2]$ und $a_i(median/2, r)$ zerlegt, wobei $median$ den Median bezeichnet.

Bei einem Itemsplit müssen alle Knoten im Baum, welche dieses Item repräsentieren ebenfalls gesplittet werden. Sind unterhalb eines Knotens, welcher ein Item repräsentiert das gesplitted wird, weitere Knoten, so werden diese Teilbäume kopiert und als neue Teilbäume in die neu entstehenden Knoten eingebunden. Die Häufigkeit des zu splittenden Items wird beim Überführen in die neuen Items halbiert (die genaue Verteilung der Daten in den einzelnen Knoten ist unbekannt, weswegen der Einfachheit halber ein Gleichverteilung angenommen wird). Ein Beispiel soll den Itemsplit verdeutlichen.

Beispiel 2 *Die Häufigkeit des Items $\langle a_1(20, 23]\rangle$ aus dem vorigen Beispiel genügt den definierten Bedingungen. Beim Itemsplit auf Basis der Intervall-Halbierung wird das Item in die beiden Subitems $\langle a_1(20, 21.5]\rangle$ und $\langle a_1(21.5, 23]\rangle$ geteilt. Es entstehen somit die beiden Itemsets $\{\langle a_1(20, 21.5]\rangle, \langle a_2(50, 100]\rangle\}$ und $\{\langle a_1(21.5, 23]\rangle, \langle a_2(20, 100]\rangle\}$ welche jeweils eine geschätzte Häufigkeit von 2 besitzen. Abbildung 4 zeigt den entsprechenden FP^2-Stream aus Abbildung 3 nach dem Split des Items $\langle a_1(20, 23]\rangle$.*

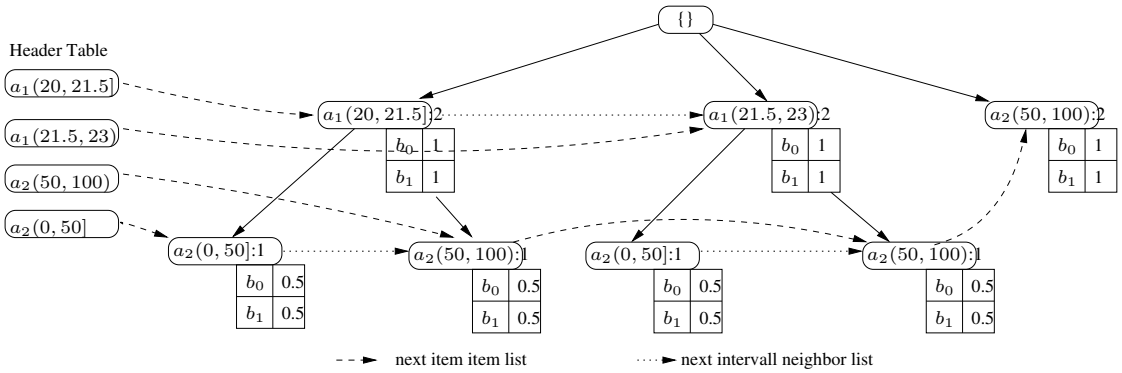


Abbildung 4: Split node $a_1(20, 23]$

Eine beliebige Verfeinerung der Intervalle ist aus Gründen einer effizienten Datenhaltung nicht sinnvoll. Somit ergibt sich die Fragestellung inwieweit Intervalle verfeinert werden. In [11] präsentieren die Autoren eine Untersuchung zur Anzahl an maximal notwendige Partitionen (Basisintervallen), welche für die anschließende Rekombination zu erzeugen sind und dabei einen möglichst geringen Informationsverlust aufweisen. Als Maß für den durch die Generalisierung entstehenden Informationsverlust führen sie die *Partial-Completeness* ein. Als K-Partial-Completeness wird dabei der maximale Support bezeichnet (als K-vielfaches des minimalen Support), den ein Itemset aufweisen darf. Basierend auf der Partial-Completeness haben die Autoren gezeigt, dass im Fall einer Partitionierung in Basisintervalle gleicher Größe, maximal $\frac{2 \cdot \alpha}{minsupp \cdot (K-1)}$. Intervalle notwendig sind. α bezeichnet dabei die Anzahl an quantitativen Attributen. Daraus lässt sich die minimale Intervallgröße für jedes Datenstromattribut a_i ableiten. Zum Zeitpunkt des ersten Erstellens des FP^2-Stream (Batch b_0) wird für jedes Attribut $a_i(min, max)$ bestimmt. Damit ergibt sich die minimale Intervallgröße $minIntSize$ entsprechend

$$minIntSize = \frac{|max - min|}{\frac{2 \cdot \alpha}{minsupp \cdot (K-1)}}. \quad (2)$$

Beim späteren Eintreffen von Transaktionen mit Werten, welche das rechts- bzw. linkseitige Extrema erweitern muss das minimale Intervall entsprechend angepasst werden.

Zusätzlich zu den Knoten müssen auch die in den Knoten eingebetteten Fenster angepasst werden. Jeder der beiden durch den Split neu entstandenen Knoten erhält hierzu die Hälfte der Häufigkeitswerte des Original-Items. Um später beim FP^2-Growth auf einfache Weise das bisherige Split-Verhalten von Items nachvollziehen zu können wird bei einem Split in jedem Zeitfensterslot ein Verweis auf das Fenster angelegt, das bei dem Split abgeteilt wurde. Um weitere Splits bzw. Merges möglichst effizient gestalten zu können verweist das letzte Fenster in der Liste auf das erste Listenelement, so dass ein Ring entsteht. Der Ring ermöglicht anschließend eine schnelle Suche auch ohne die Verwendung einer doppelt verketteten Liste.

4.4 Item-Merge

Items, welche nahezu die gleiche Informationsdichte aufweisen, können ohne weiteren Informationsverlust kombiniert (generalisiert) werden. Die Generalisierung von Items führt somit zu einer kompakteren Datenstruktur.

Zwei Items im FP^2-Stream $a_i(l, r)$ und $a_i(l', r')$ können genau dann zu dem Item $a_i(l, r')$ zusammengefasst werden, wenn für alle Knoten die diese Items repräsentieren folgenden Bedingungen erfüllt werden:

1. Die Items $a_i(l, r)$ und $a_i(l', r')$ sind *direkte Nachbarn*, d.h. es gilt $r = l'$.
2. Benachbarte Knoten, welche die Items $a_i(l, r)$ und $a_i(l', r')$ repräsentieren besitzen den gleichen Präfix, d.h. sie verfügen über den gleichen Elternknoten im FP^2-Stream.
3. $density(a_i(l, r)) \sim density(a_i(l', r'))$

1. stellt sicher, dass es sich bei den Verbundkandidaten um Intervallnachbarn handelt. Ausschließlich direkte Nachbarn können miteinander verbunden werden. 2. garantiert, dass alle von dem Item-Merge betroffenen Itemsets ebenfalls verbunden werden können. 3. stellt sicher, dass die zu verbindenden Items über nahezu den gleichen Informationsgehalt verfügen, d.h. das es durch die Generalisierung zu keinem Informationsverlust kommt.

Algorithmus 1 führt den Merge umgehend nach dem Split aus. Wurde ein Split auf einem Item durchgeführt, dann verfügen die beiden resultierenden Items über die gleiche Dichte. Erst durch das Einfügen neuer Batches setzt sich die Schiefe, welche zuvor für den Split notwendigerweise festgestellt wurde, auch in den neuen Items durch. Aus diesem Grund ist eine sofortige Rekombination zuvor erst geteilter Items für die nächsten k Batches nicht sinnvoll.

4.5 Prunning und Reorganisation

Durch das Splitten von Items wird der FP^2-Stream kontinuierlich vergrößert. So resultiert zum Beispiel der Split des Wurzelknotens in einer Verdoppelung aller Knoten im Baum. Um diesen dennoch möglichst klein und damit die Verarbeitung effizient zu gestalten werden zwei verschiedene Ansätze verfolgt:

- (i) Itemsets, welche nicht mehr häufig sind und in den nächsten Zeitschritten auch nicht mehr häufig werden können, werden aus der Datenstruktur entfernt werden.
- (ii) Es erfolgt eine Reorganisation der Baumstruktur um eine möglichst hohe Kompaktheit zu gewähren (ähnlich dem CPT [12]).

Item-Rekonstruktion und Itemset Exktraktion

Zwar werden im FP^2-Stream häufige Itemsets in einer kompakten Darstellung gespeichert, es wird aber nicht garantiert, dass die Extraktion der häufigen Muster effizient geschieht. Das Herauslösen der Frequent Pattern stellt dabei ein kombinatorisches Problem dar. Mit der Entwick-

lung des FP-Tee präsentierten Han et al. in [5] den FP-Growth, ein Verfahren zur Extraktion aller im FP-Tree gespeicherten häufigen Muster. Im Folgenden wird ein für den FP^2-Stream angepasstes Growth-Verfahren zum extrahieren der häufigen Muster beschrieben.

Bevor die häufigen Itemsets extrahiert werden können sind zwei wesentliche Verarbeitungsschritte notwendig. In einem ersten Schritt wird aus dem FP^2-Stream ein FP^2-Tree extrahiert. Hierzu wird die Summe über alle in dem Fenster eines Knotens enthaltenen Häufigkeiten als aktueller Wert für einen Knoten des FP^2-Tree herangezogen. Aufgrund des kontinuierlichen Teilens der Knoten, unter Annahme einer Gleichverteilung der einzelnen Werte in den Intervallen, handelt es sich bei den Häufigkeiten in den Knoten des FP^2-Stream zumeist nur um approximierte Werte. Lediglich Knoten, welche zuvor nicht gesplittet wurden, weißen genaue Häufigkeitswerte auf. Um dies auch für zuvor geteilte Knoten zu erreichen sind die Fensterslots aller Knoten, welche ursprünglich von dem selben Knoten abstammen untereinander über Ringlisten miteinander verbunden (siehe Item-Split). Unter Verwendung dieser kann die genaue Häufigkeit für den ursprünglichen Knoten wiederhergestellt werden (die Summe über alle Knoten einer Ringliste).

Der durch den Extraktionsprozess erzeugte FP^2-Tree enthält im allgemeinen quantitative Items, deren Support nicht $minsupp$ genügt. Im nächsten Schritt werden durch die Rekombination von benachbarten Intervallen Items erzeugt, welche $minsupp$ genügen. DIn der Literatur lassen sich verschiedene Interessantheitsmaße für das Erzeugen der Items finden [10; 13]. Beim FP^2-Stream bzw. FP^2-Tree findet die Informationsdichte der Items Verwendung. D.h. das Item mit der höchsten Informationsdichte wird so lange mit benachbarten Items (dem jeweils dichtesten direkt benachbarten Item) verbunden, bis der geforderte minimale Support erreicht wird. Werden per Definition mehr als ein häufiges Item pro Datenstromattribut gefordert, dann wird dieses Verfahren auf die verbliebenen nicht-häufigen Items angewandt.

Nachdem der FP^2-Tree erzeugt wurde und die Items den definierten Bedingungen genügen, können aus diesem häufige Itemsets extrahiert werden. Aus Gründen der effizienten Auswertung wird hierzu auf das *Top-Down* Verfahren nach [14] zurückgegriffen.

5 Evaluierung

Im folgenden soll die Funktionsweise des vorgestellten Verfahrens gezeigt werden. Zu diesem Zweck wurde das Verfahren in Form eines Operators in das Datenstrom-Managementsystem AnduIN [6] integriert. Neben einer Vielzahl an einfachen Operationen (Filter, Projektion, Verbund, Aggregation) unterstützt AnduIN zusätzlich auch die Integration komplexer Operatoren in Form von Synopsen-Operatoren. Unter einem Synopsen-Operator wird dabei ein Operator verstanden, welcher sowohl über eine In-Memory Datenzusammenfassung, als auch über notwendige Algorithmen für das Einfügen von Daten und deren Auswertung verfügt. Das hier präsentierte Verfahren wurde als ein solcher Synopsenoperator integriert.

Zunächst soll die grundlegende Funktion des Verfahrens präsentiert werden. Hierzu wurden 3 Datenströme zu je 5000 Datenpunkten erzeugt. Ein Datenstrom entspricht dabei den Daten eines Attributes, d.h. die Analyse erfolgt im weiteren über Attributen. Die erzeugten Werte sind dabei standardnormalverteilt. Abbildung 5(a) zeigt exemplarisch den erzeugten Datenstrom für ein Attribut.

Aus diesen drei Datenströmen sollen nun zusammenhängende Itemsets extrahiert werden, wobei pro Attribut ein Item erzeugt werden soll. Im ersten Test sollen Items bzw. Itemsets extrahiert werden, deren Support $minsupp$ mindesten 0.25 ist. Es wurde mit einer Batchgröße $|b|$ = 100, einer Fenstergröße N von 5, sowie Histogrammen mit 10 Buckets d und einer maximalen Schiefe $maxskew$ von 0.2 getestet. Abbildung 5(b) zeigt beispielhaft die zeitliche Entwicklung des über dem Datenstrom aus Abbildung 5(a) entwickelten Items. Graue Buckets repräsentieren Items (pro Zeitschritt jeweils eine zusammenhängende Box). Die schwarzen Rahmen entsprechen den Knoten im FP^2-Stream, die über diesem Attribut existieren. Weiß gerahmte Buckets sind somit Knoten im FP^2-Stream, welche nicht häufig sind. Graue Balken, die aus mehreren Buckets bestehen sind im FP^2-Stream über mehrere Knoten verteilt.

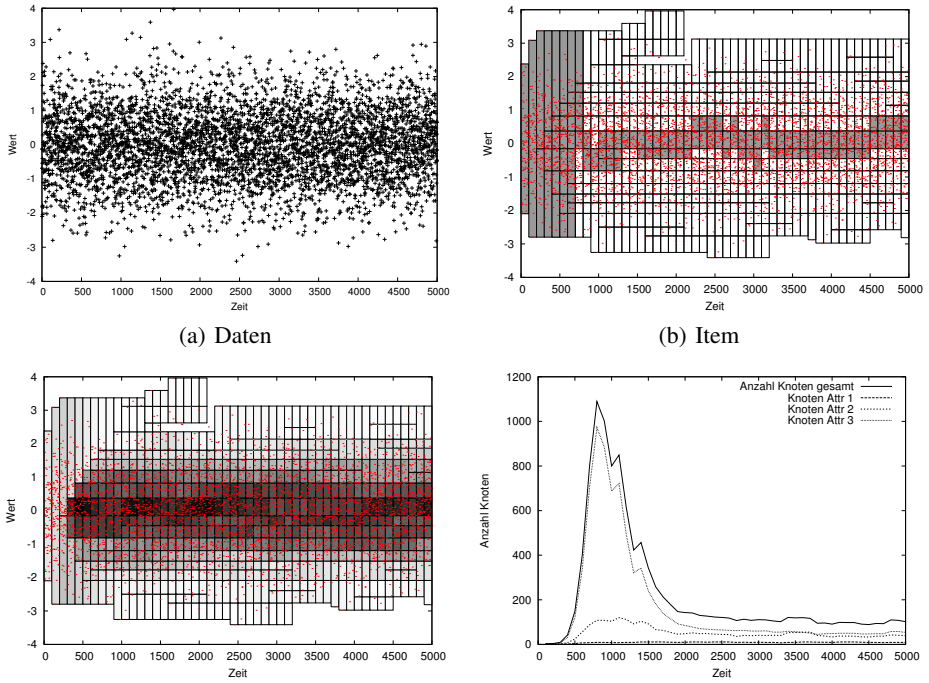
Abbildung 5(b) zeigt sehr gut, dass zu Beginn der Analyse der vollständige Wertebereich durch das Item überdeckt wird. Anschließend fängt der Algorithmus an, den alles überdeckenden Knoten in mehrere Teilknoten zu zerlegen. Aufgrund des Zeitfensters von 5 Batches setzt sich diese Verfeinerung allerdings erst nach 600 Zeiteinheiten durch. Anschließend schwankt das erzeugte Item um den Mittelwert der Standardnormalverteilten Daten.

Elementar für die Menge an Knoten im FP^2-Stream und für die Konstruktion der Items bzw. Itemsets ist das Maß der Informationsdichte. Abbildung 5(c) zeigt für den betrachteten Beispieldatenstrom die Dichte der einzelnen Buckets. Die Bucketgrenzen entsprechen denen aus Abbildung 5(b). Die Abbildung zeigt die mit steigender Knotenverfeinerung zunehmende Informationsdichte.

Die Synopse des FP^2-Stream ist eine Datenstruktur zur effizienten Speicherung von Transaktionen. Das kontinuierliche Verfeinern von Knoten hat allerdings starken Einfluss auf die Größe der Baumstruktur. So führt zum Beispiel ein Splitt in der Wurzel zu einer Verdoppelung aller abhängigen Knoten. Während der initialen Phase muss sich das Verfahren erst an die Daten anpassen. Dies kann eine Vielzahl an Split und Merge Operationen zur Folge haben. In Abbildung 5(d) ist die Zeitliche Entwicklung zu obigen Beispiel dargestellt. Sehr gut zu erkennen ist die Spitze zu Beginn des Verarbeitungsprozesses.

Abbildung 5(d) zeigt außerdem die Anzahl an Knoten pro Attribut im Baum. Erwartungsgemäß steigt diese Zahl mit der Tiefe des Baumes, so dass das Attribut, welches sich auf Blattebene befindet (im Beispiel Attribut 3) durch die höchste Anzahl an Knoten repräsentiert wird.

Durch das Rekombinieren von Items innerhalb des FP^2-Stream werden automatisch auch die entsprechenden Itemsets zusammengeführt. Während der Itemset-Exktraktion werden alle Items aus der Datenstruktur herausgelöst, welche dem minimalen Support genügen. Im Beispiel der standardnormalverteilten Daten sind hierbei zusätzlich 2-Itemsets bzw. vereinzelt auch 3-Itemsets extrahiert wurden. Die Bilder in Abbildung 6 zeigen die zeitliche Entwicklung des 2-Itemsets über den Attributen 0 und 2. Jedes der einzelnen Bilder entspricht dabei dem am Ende eines Batches extrahiertem Itemset. Wiederum sehr gut zu erkennen ist die initiale Phase. Außerdem sehr gut zu erkennen ist das entfernen von Buckets am Rand infolge zu weniger Werte in diesen Regionen. Das entfernen von uninteressanten Knoten ist einer der Gründe, warum sich die Anzahl an Knoten im FP^2-Stream nach dem initialen Anstieg auf ei-



(a) Daten (b) Item

(c) Dichteverteilung (d) Anzahl Knoten

Abbildung 5: Standardnormalverteilte Daten

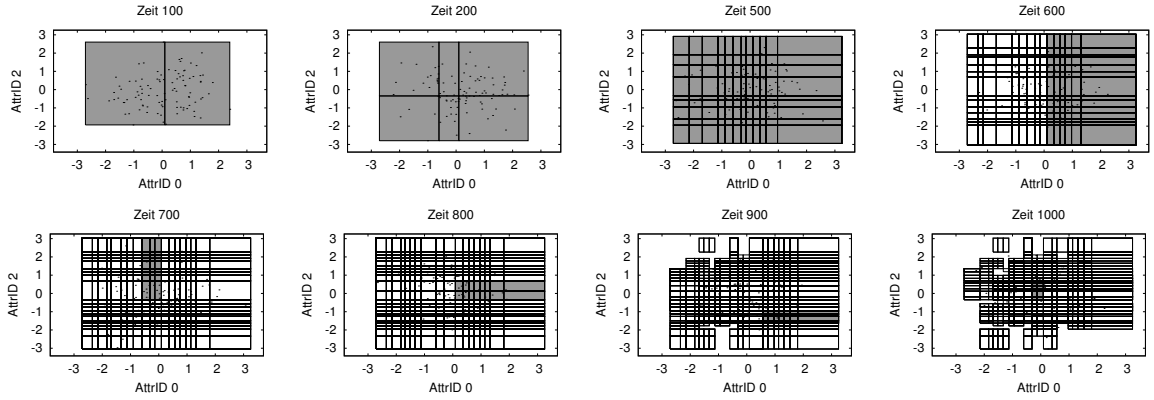


Abbildung 6: Entwicklung des Itemsets s1 und s2

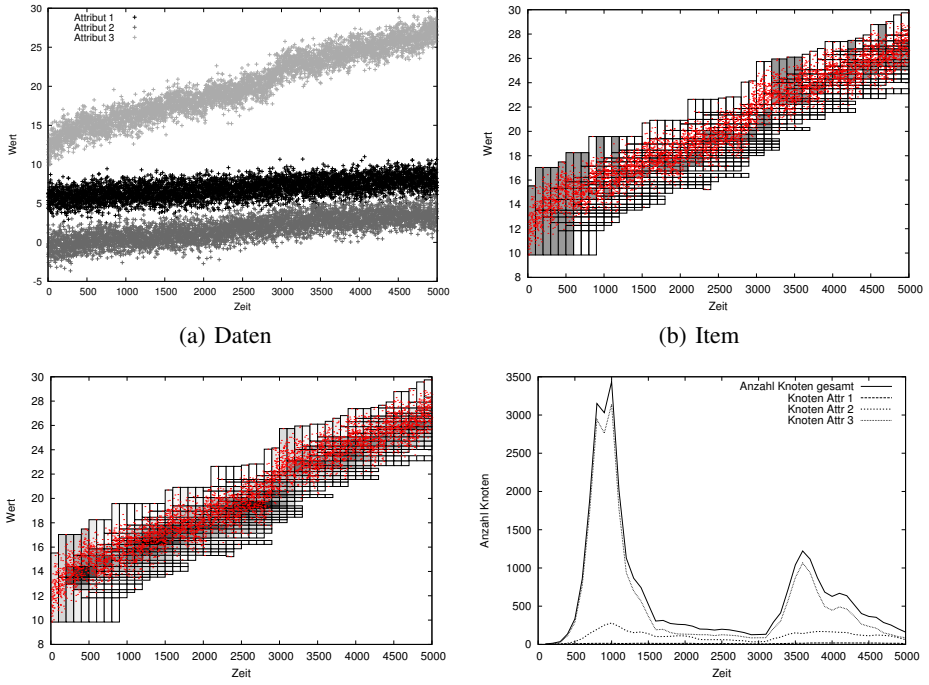
nem deutlich niedrigeren Level einpendelt.

Für den nächsten Test wurden wiederum drei Datenströme erzeugt. Jeder Datenstrom umfasst dabei 5000 Datenpunkte und folgt einem Trend. Zusätzlich wurden die Daten mit normalverteiltem Rauschen überlagert. Abbildung 7(a) zeigt den zeitlichen Verlauf der drei Datensätze.

Das Experiment wurde wiederum mit $|b| = 100$, $N = 5$ und einer maximalen Schiefe $maxskew$ von 0.2 bei 10 Histogrammbuckets durchgeführt. Der geforderte Support betrug 0.15. Abbildung 7(b) zeigt exemplarisch die Entwicklung der Items für ein Attribut. Die notwendige Einschwingphase ist deutlich zu erkennen. Im weiteren Verlauf entwickelt sich das Item mit dem Trend, wobei für den geforderten minimalen Support mehr oder weniger viele Knoten kombiniert werden müssen. Interessant ist die Entwicklung des Items um den Zeitpunkt 1000. Hier sorgen die neu eintreffenden Werte offensichtlich für ein oszillieren zwischen zwei Items, welches sich erst mit dem Teitpunkt 1300 durchsetzen kann.

In Abbildung 7(c) ist die Informationsdichte-Entwicklung des entsprechenden Attributes dargestellt und Abbildung 7(d) zeigt zeitliche Entwicklung der Knotenanzahl im FP^2-Stream. Trotz des kontinuierlichen Trends entwickelt sich die Gesamtknotenanzahl anschließend relativ konstant bei ca. 200 Knoten. Beginnend ab Zeitpunkt 3300 wächst die Knotenanzahl erneut drastisch an. Die Ursache hierfür liegt offensichtlich in einer Änderung der Datencharakteristik während dieser Zeit bei Attribut 3. Trotz dem, dass es sich um trenbehaftet Daten mit normalverteiltem Rauschen handelt, scheint zwischen den Zeitpunkten 3000 und 3500 eine weitere Überlagerung (ähnlich einem Burst) aufzutreten. Diese plötzliche Veränderung hat entsprechend Auswirkungen auf die Knoten und deren Dichten (siehe Abbildung 7(c)), welche Attribut 3 repräsentieren und führen vermutlich zu dem Peak.

Diese ersten Ergebnisse zeigen, dass das Verfahren des quantitativen FP-Mining für die Analyse von Datenströmen prinzipiell geeignet ist. Es wurde gezeigt, dass das eingeführte Maß der Informationsdichte ist für Erzeugen in-



(a) Daten (b) Item

(c) Dichteverteilung (d) Anzahl Knoten

Abbildung 7: Trendbehaftete Daten

teressanter Itemsets. Erst dieses ermöglicht die Analyse über Datenströmen, da es sowohl für das Teilen und Verbinden von Intervallen, als auch für die Rekombination als Schritt der Itemsetextraktion herangezogen wird. Mit dem präsentierten Maß kann auch das Testen aller möglichen Intervallkombinationen verzichtet werden, was einer der wesentlichen Gründe für die Eignung zur Analyse von Datenströmen ist.

6 Zusammenfassung

Das FP-Mining über statischen Daten mit kategorischen Werte zählt heute zu den klassischen Data Mining Verfahren, für welches eine Vielzahl an Lösungen in der Literatur zu finden sind. In der hier vorgestellten Arbeit wurde ein neuer Ansatz für die Identifikation von häufigen Mustern über quantitative Attribute aus Datenströmen präsentiert. Das präsentierte Verfahren kombiniert hierzu bekannte Techniken aus dem Data Mining Bereich mit Verfahren der mehrdimensionalen Indexstrukturen. Neben den notwendigen Verarbeitungsschritten wurde die prinzipielle Funktion anhand von Beispielen gezeigt. In weiteren Arbeiten soll die Funktion des Verfahrens mit realen Szenarien untersucht und evaluiert werden. Das DSMS AnduIN erlaubt die Teilweise Auslagerung von Funktionalität in Wireless Sensor Netzwerke. Eine der zukünftigen Arbeiten ist daher die Migration des vorgestellten Verfahrens hin zu einem In-Network-Operator von AnduIN mit dem Ziel einer möglichst effizienten Verarbeitung.

Literatur

[1] R. Agrawal, T. Imielinski, and A. N. Swami. Mining Association Rules between Sets of Items in Large Databases. In *'93, Washington, D.C., USA*, pages 207–216, 1993.

[2] Y. Aumann and Y. Lindell. A statistical theory for quantitative association rules. In *Journal of Intelligent Information Systems*, pages 261–270, 1999.

[3] T. Fukuda, Y. Morimoto, Sh. Morishita, and T. Tokuyama. Mining optimized association rules for numeric attributes. In *PODS '96*, pages 182–191. ACM, 1996.

[4] M. Halatchev and Le Gruenwald. Estimating missing values in related sensor data streams. In *COMAD*, pages 83–94, 2005.

[5] J. Han, J. Pei, and Y. Yin. Mining Frequent Patterns without Candidate Generation. In *2000, Dallas, USA*, pages 1–12, 2000.

[6] D. Klan, K. Hose, M. Karnstedt, and K. Sattler. Power-aware data analysis in sensor networks. In *ICDE '10*, Long Beach, California, USA, 2010. IEEE, IEEE.

[7] C. K.-S. Leung and Q. I. Khan. Dstree: A tree structure for the mining of frequent sets from data streams. In *ICDM '06*, pages 928–932, Washington, DC, USA, 2006. IEEE Computer Society.

[8] G. S. Manku and R. Motwani. Approximate Frequency Counts over Data Streams. In *2002, Hong Kong, China*, pages 346–357, 2002.

[9] J. S. Park, M.-S. Chen, and P. S. Yu. An effective hash-based algorithm for mining association rules. In *SIGMOD '95*, pages 175–186, New York, NY, USA, 1995. ACM.

[10] G. Piatetsky-Shapiro. *Discovery, analysis and presentation of strong rules*. 1991.

[11] R. Srikant and R. Agrawal. Mining quantitative association rules in large relational tables. *SIGMOD Rec.*, 25(2):1–12, 1996.

[12] S. K. Tanbeer, Ch. F. Ahmed, B.-S. Jeong, and Y.-K. Lee. Efficient frequent pattern mining over data streams. In *CIKM '08*, pages 1447–1448, New York, NY, USA, 2008. ACM.

[13] A. Tuzhilin. On subjective measures of interestingness in knowledge discovery. In *KDD '95*, pages 275–281. AAAI Press, 1995.

[14] K. Wang, L. Tang, J. Han, and J. Liu. Top down fp-growth for association rule mining. In *PAKDD '02*, pages 334–340, London, UK, 2002. Springer-Verlag.

A Novel Multidimensional Framework for Evaluating Recommender Systems

Artus Krohn-Grimberghe
Information Systems and Machine Learning Lab
University of Hildesheim, Germany
artus@ismll.de

Alexandros Nanopoulos
Information Systems and Machine Learning Lab
University of Hildesheim, Germany
nanopoulos@ismll.de

Lars Schmidt-Thieme
Information Systems and Machine Learning Lab
University of Hildesheim, Germany
schmidt-thieme@ismll.de

ABSTRACT

The popularity of recommender systems has led to a large variety of their application. This, however, makes their evaluation a challenging problem, because different and often contrasting criteria are established, such as accuracy, robustness, and scalability. In related research, usually only condensed numeric scores such as RMSE or AUC or F-measure are used for evaluation of an algorithm on a given data set. It is obvious that these scores are insufficient to measure user satisfaction.

Focussing on the requirements of business and research users, this work proposes a novel, extensible framework for the evaluation of recommender systems. In order to ease user-driven analysis we have chosen a multidimensional approach. The research framework advocates interactive visual analysis, which allows easy refining and reshaping of queries. Integrated actions such as drill-down or slice/dice, enable the user to assess the performance of recommendations in terms of business criteria such as increase in revenue, accuracy, prediction error, coverage and more.

The ability of the proposed framework to comprise an effective way for evaluating recommender systems in a business-user-centric way is shown by experimental results using a research prototype.

Keywords

Recommender Systems, Recommendation, Multidimensional Analysis, OLAP, Exploratory Data Analysis, Performance Analysis, Data Warehouse

1. INTRODUCTION

The popularity of recommender systems has resulted in a large variety of their applications, ranging from presenting personalized web-search results over identifying preferred multimedia content (movies, songs) to discovering friends in social networking sites. This broad range of applications, however, makes the evaluation of recommender systems a challenging problem. The reason is the different and often contrasting criteria that are being involved in real-world applications of recommender systems, such as their accuracy, robustness, and scalability.

The vast majority of related research usually evaluates recommender system algorithms with condensed numeric scores: root mean square error (RMSE) or mean absolute error (MAE) for rating prediction, or measures usually stemming from information retrieval such as precision/recall or F-measure for item prediction. Evidently, although such measures can indicate the performance of algorithms regarding some perspectives of recommender systems' applications, they are insufficient to cover the whole spectrum of aspects involved in most real-world applications. As an alternative approach towards characterizing user experience as a whole, several studies employ user-based evaluations. These studies, though, are usually rather costly, difficult in design and implementation.

More importantly, when recommender systems are deployed in real-world applications, notably e-commerce, their evaluation should be done by business analysts and not necessarily by recommender-system researchers. Thus, the evaluation should be flexible on testing recommender algorithms according to business analysts' needs using interactive queries and parameters. What is, therefore, required is to provide support for evaluation of recommender systems' performance based on popular online analytical processing (OLAP) operations. Combined with support for visual analysis, actions such as drill-down or slice/dice, allow assessment of the performance of recommendations in terms of business objectives. For instance, business analysts may want to examine various performance measures at different levels (e.g., hierarchies in categories of recommended products), detect trends in time (e.g., elevation of average product rating following a change in the user interface), or segment the customers and identify the recommendation quality with respect to each customer group. Furthermore, the interactive and visual nature of this process allows easy adaptation of the queries according to insights already gained.

In this paper, we propose a novel approach to the evaluation of recommender systems. Based on the aforementioned motivation factors, the proposed methodology builds on multidimensional analysis, allowing the consideration of various aspects important for judging the quality of a recommender system in terms of real-world applications. We describe a way for designing and developing the proposed extensible multidimensional framework, and provide insights

into its applications. This enables integration, combination and comparison of both, the presented and additional, measures (metrics).

To assess the benefits of the proposed framework, we have implemented a research prototype and now present experimental results that demonstrate its effectiveness.

Our main contributions are summarized as follows:

- A flexible multidimensional framework for evaluating recommender systems.
- A comprehensive procedure for efficient development of the framework in order to support analysis of both, dataset facets and algorithms' performance using interactive OLAP queries (e.g., drill-down, slice, dice).
- The consideration of an extended set of evaluation measures, compared to standards such as the RMSE.
- Experimental results with intuitive outcomes based on swift visual analysis.

2. RELATED WORK

For general analysis of recommender systems, Breese [5] and Herlocker et al. [11] provide a comprehensive overview of evaluation measures with the aim of establishing comparability between recommender algorithms. Nowadays, the generally employed measures within the prevailing recommender tasks are MAE, (R)MSE, precision, recall, and F-measure. In addition further measures including confidence, coverage and diversity related measures are discussed but not yet broadly used. Especially the latter two have attracted attention over the last years as it is still not certain whether today's predictive accuracy or precision and recall related measures correlate directly with interestingness for a system's end users. As such various authors proposed and argued for new evaluation measures [22, 21, 6]. Ziegler [22] has analyzed the effect of diversity with respect to user satisfaction and introduced topic diversification and intra-list similarity as concepts for the recommender system community. Zhang and Hurley [21] have improved the intra-list similarity and suggested several solution strategies to the diversity problem. Celma and Herrera [6] have addressed the closely related novelty problem and propose several technical measures for coverage and similarity of item recommendation lists. All these important contributions focus on reporting single aggregate numbers per dataset and algorithm. While our framework can deliver those, too, it goes beyond that by its capability of combining the available measures and, most importantly, dissecting them among one or more dimensions.

Analysis of the end users' response to recommendations and their responses' correlation with the error measures used in research belongs to the field of Human-Recommender Interaction. It is best explored by user studies and large scale experiments, but both are very expensive to obtain and thus rarely conducted and rather small in scale. Select studies are [13, 14, 4]. Though in the context of classical information retrieval, Joachims et al [13] have conducted a highly relevant study on the biasing effect of the position an item has within a ranked list. In the context of implicit feedback vs. explicit feedback Jones et al [14] have conducted an important experiment on the preferences of users concerning recommendations generated by unobtrusively collected implicit feedback compared to recommendations based on explicitly stated preferences. Bollen et al. [4] have researched the effect of recommendation list length in combination with recommendation quality on perceived choice satisfaction. They found that for high quality recommendations, longer lists tend to overburden the user with difficult choice decisions. Against the background of those results we believe that for initial research on a dataset, forming an idea, checking if certain effects are present, working on collected data with a framework like the one presented is an acceptable proxy. With findings gained in this process, conducting meaningful user studies is an obvious next step.

Recent interesting findings with respect to dataset characteristics are e.g. the results obtained during the Netflix challenge [3, 17] on user and item base- effects and time-effects in data. When modeled appropriately, they have a noteworthy effect on recommender performance. The long time it took to observe these properties of the dataset might be an indicator for the fact that with currently available tools proper analysis of the data at hand is more difficult and tedious than it should be. This motivates the creation of easy-to-use tools enabling thorough analysis of the datasets and the recommender algorithm's results and presenting results in an easy to consume way for the respective analysts.

Notable work regarding the integration of OLAP and recommender systems stems from the research of Adomavicius et al. [2, 1]. They treat the recommender problem setting with its common dimensions of users, items, and rating as inherently multidimensional. But unlike this work, they focus on the multidimensionality of the generation of recommendations and on the recommenders themselves being multidimensional entities that can be queried like OLAP cubes (with a specifically derived query language, RQL). In contrast, our work acknowledges the multidimensional nature of recommender systems, but focusses on their multidimensional evaluation.

Existing frameworks for recommender systems analysis usually focus on the automatic selection of one recommendation technique over another. E.g., [10] is focussed on an API that allows retrieval and derivation of user satisfaction with respect to the recommenders employed. The AWESOME system by Thor and Rahm [20], the closest approach to that presented here, shares the data warehousing approach, the description of the necessary data preparation (ETL), and the insight of breaking down the measures used for recommender performance analysis by appropriate categories. But contrary to the approach presented here, the AWESOME framework is solely focussed on website performance and relies on static SQL-generated reports and decision criteria. Furthermore, it incorporates no multidimensional approach and does not aim at simplifying end-user-centric analysis or interactive analysis at all.

3. FRAMEWORK REQUIREMENTS

3.1 The Role of a Multidimensional Model

Business analysts expect all data of a recommender systems (information about items, generated recommendations, user preferences, etc.) to be organized around business entities in form of dimensions and measures based on a multidimensional model. A multidimensional model enforces structure upon data and expresses relationships between data elements [19]. Such a model, thus, allows business analysts

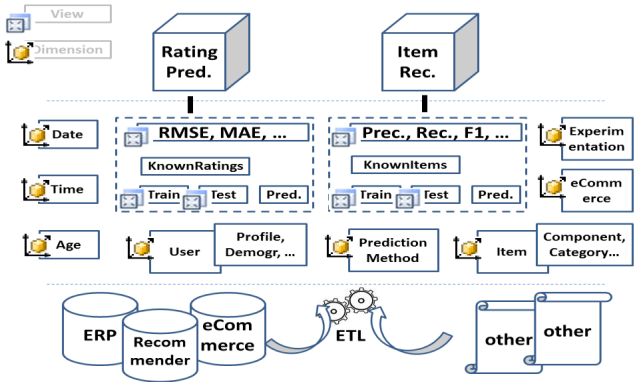


Figure 1: The recommender evaluation framework. The dimensions specified are connected with both fact table groups (dashed boxes in the center) and are thus available in both resulting cubes. End users can connect to the Rating Prediction and Item Recommendation cubes.

to investigate all aspects of their recommender system by using the popular OLAP technology [7]. This technology provides powerful analytical capabilities that business analysts can query to detect trends, patterns and anomalies within the modeled measures of recommender systems' performance across all involved dimensions.

Multidimensional modeling provides comprehensibility for the business analysts by organizing entities and attributes of their recommender systems in a parent-child relationship (1:N in databases terminology), into dimensions that are identified by a set of attributes. For instance, the dimension of recommended items may have as attributes the name of the product, its type, its brand and category, etc. For the business analyst, the attributes of a dimension represent a specific business view on the facts (or key performance indicators), which are derived from the intersection entities. The attributes of a dimension can be organized in a hierarchical way. For the example of a dimension about the user of the recommender systems, such a hierarchy can result from the geographic location of the user (e.g., address, city, or country). In a multidimensional model, the measures (sometimes called facts) are based in the center with the dimensions surrounding them, which forms the so called star schema that can be easily recognized by the business analysts. The star schema of the proposed framework will be analyzed in the following section.

It is important to notice that aggregated scores, such as the RMSE, are naturally supported. Nevertheless, the power of a multidimensional model resides in adding further derived measures and the capability of breaking all measures down along the dimensions defined in a very intuitive and highly automated way.

3.2 Core Features

Organizing recommender data in a principled way provides automation and tool support. The presented framework enables analysis of all common recommender datasets. It supports both rating prediction and item recommendation scenarios. Besides that, data from other application sources can and should be integrated for enriched analysis capabilities. Notable sources are ERP systems, eCommerce systems and experimentation platform systems employing recommender systems. Their integration leverages analysis of the recommender data by the information available within the application (e.g., recommender performance given the respective website layouts) and also analysis of the application data by recommender information (e.g., revenue by recommender algorithm).

Compared to RMSE, MAE, precision, recall, and F-measure, more information can be obtained with this framework as, first, additional measures e.g. for coverage, novelty, diversity analysis are easily integrated and thus available for all datasets. Second, all measures are enhanced by the respective ranks, (running) differences, (running) percentages, totals, standard deviations and more.

While a single numerical score assigned to each recommender algorithm's predictions is crucial for determining winners in challenges or when choosing which algorithm to deploy [8], from an business insight point of view a lot of interesting information is forgone this way. Relationships between aspects of the data and their influence on the measure may be hidden. One such may be deteriorating increase in algorithmic performance with respect to an increasing number of rating available per item, another the development of the average rating over the lifetime of an item in the product catalog. A key capability of this framework is exposing intuitive ways for analyzing the above measures by other measures or related dimensions.

From a usability point of view, this framework contributes convenient visual analysis empowering drag-drop analysis and interactive behavior. Furthermore, convenient visual presentation of the obtained results is integrated from the start as any standard conforming client can handle it. Manual querying is still possible as is extending the capabilities of the framework with custom measures, dimensions, or functions and post-processing of received results in other applications. Inspection of the original source data is possible via custom actions which allow the retrieval of the source rows that produced the respective result. Last but not least, aggregations allow for very fast analysis of very large datasets, compared to other tools.

The following section elaborates on the architecture of the multidimensional model that is used by the proposed framework, by providing its dimensions and measures.

4. THE ARCHITECTURE OF THE MULTIDIMENSIONAL FRAMEWORK

Figure 1 gives an overview of the architecture of the framework. The source data and the extract-transform-load (ETL) process cleaning it and moving it into the data store are located at the bottom of the framework. The middle tier stores the collected information in a data warehouse manner regarding facts (dashed boxes in the center) and dimensions (surrounding the facts). The multidimensional cubes (for rating recommendation and item prediction) sitting on top of the data store provide access to an extended set of measures (derived from the facts in the warehouse) that allow automatic navigation along their dimensions and interaction with other measures.

4.1 The Data Flow

The data gathered for analysis can be roughly divided into two categories:

***Core data*:** consisting of the algorithms' training data, such as past ratings, purchase transaction information, online click streams, audio listening data, ... and the persisted algorithms' predictions.

***Increase-insight data*:** can be used as a means to leverage the analytic power of the framework. It consists roughly of user master data, item master data, user transactional statistics, and item transactional statistics. This data basically captures the metadata and usage statistics data not directly employed by current recommender algorithms (such as demographic data, geographic data, customer performance data...).

In case of recommender algorithms employed in production environments, relational databases housing the transactional system (maybe driving an e-commerce system like an ERP system or an online shop) will store rich business master data such as item and user demographic information, lifetime information and more, next to rating information, purchase information, and algorithm predictions. In case of scientific applications, different text files containing e.g. rating information, implicit feedback, and the respective user and item attributes for training and the algorithms' predictions are the traditional source of the data.

From the respective source, the master data, the transactional data, and the algorithm predictions are cleaned, transformed, and subsequently imported into a data warehouse. Referential integrity between the elements is maintained, so that e.g. ratings to items not existing in the system are impossible. Incongruent data is spotted during insert into the recommender warehouse and presented to the data expert.

Inside the framework, the data is logically split into two categories: measures (facts) that form the numeric information for analysis, and dimensions that form the axes of analysis for the related measures. In the framework schema (figure 1), the measures are stylized within the dashed boxes. The dimensions surrounding them and are connected to both, the rating prediction and the item recommendation measures.

4.2 The Measures

Both groups of measures analyzed by the framework—the measures for item recommendation algorithms and the measures for rating prediction algorithms—can be divided into basic statistical and information retrieval measures.

***Statistical measures*:** Among the basic statistical measures are counts and distinct counts, ranks, (running) differences and (running) percentages of various totals for each dimension table, train ratings, test ratings and predicted ratings; furthermore, averages and their standard deviations for the lifetime analysis, train ratings, test ratings, and predicted ratings.

***Information retrieval measures*:** Among the information retrieval measures are the popular MAE and (R)MSE for rating prediction, plus user-wise and item-wise aggregated precision, recall and F-measure for item prediction. Novelty, diversity, and coverage measures are also included as they provide additional insight. Furthermore, for comparative analysis, the differences in the measures between any two chosen (groups of) prediction methods are supported as additional measures.

In case a recommender system and thus this framework is accompanied by a commercial or scientific application, this application usually will have measures of its own. These measures can easily be integrated into the analysis. An example may be an eCommerce application adding sales measures such as gross revenue to the framework. These external measures can interact with the measures and the dimension of the framework.[1]

4.3 The Dimensions

The dimensions are used for slicing and dicing the selected measures and for drilling down from global aggregates to fine granular values. For our framework, the dimensions depicted in figure 1 are:

***Date*:** The Date dimension is one of the core dimensions for temporal analysis. It consists of standard members such as Year, Quarter, Month, Week, Day and the respective hierarchies made up from those members. Furthermore, Year-to-date (YTD) and Quarter/Month/Week/Day of Year logic provides options such as searching for a Christmas or Academy Awards related effect.

***Time*:** The Time dimension offers Hour of Day and Minute of Day/Hour analysis. For international datasets this dimension profits from data being normalized to the time zone of the creator (meaning the user giving the rating).

***Age*:** The Age dimension is used for item and user lifetime analysis. Age refers to the relative age of the user or item at the time the rating is given/received or an item from a recommendation list is put into a shopping basket and allows for analysis of trends in relative time (c.f. section 6).

***User*:** User and the related dimensions such as UserProfile and UserDemographics allow for analysis by user master data and by using dynamically derived information such as activity related attributes. This enables grouping of the users and content generated by them (purchase histories, ratings) by information such as # of ratings or purchases, # of days of activity, gender, geography...

***Item*:** Item and the related dimensions such as ItemCategory and ItemComponent parallel the user-dimensions. In a movie dataset, the item components could be, e.g., actors, directors, and other credits.

***Prediction Method*:** The Prediction Method dimension allows the OLAP user to investigate the effects of the various classes and types or recommender systems and their respective parameters. Hierarchies, such as Recommender Class, Recommender Type, Recommender Parameters, simplify the navigation of the data.

***eCommerce*:** As recommender algorithms usually accompany a commercial or scientific application (e.g., eCommerce) having dimensions of its own, these dimensions can easily be integrated into and be used by our framework.

[1] E.g., the revenue could be split up by year and recommendation method, showing the business impact of a recommender.

***Experimentation*:** In case this framework is used in an experiment-driven scenario [8], such as an online or marketing setting, Experimentation related dimensions should be used. They parallel the PredictionMethod dimension, but are more specific to their usage scenario.

5. PROTOTYPE DESCRIPTION

This section describes the implementation of a research prototype for the proposed framework. The prototype was implemented using Microsoft SQL Server 2008 [18] and was used later for our performance evaluation.

In our evaluation, the prototype considers the Movielens 1m dataset [9], which is a common benchmark for recommender systems. It consists of 6.040 users, 3.883 items, and 1.000.209 ratings received over roughly three years. Each user has at least 20 ratings and the metadata supplied for the users is userId, gender, age bucket, occupation, and zipcode. Metadata for the item is movieId, title and genre information.

Following a classical data warehouse approach [15, 12], the database tables are divided into dimension and fact tables. The dimension tables generally consist of two kinds of information: static master data and dynamic metadata. The static master data usually originates from an ERP system or another authoritative source and contains e.g. naming information. The dynamic metadata is derived information interesting for evaluation purposes, such as numbers of ratings given or time spent on the system. To allow for an always up to date and rich information at the same time, we follow the approach of using base tables for dimension master data and views for dynamic metadata derived through various calculations. Further views then expose the combined information as pseudo table. The tables used in the warehouse of the prototype are Date, Time, Genre (instantiation of Category), Item, ItemGenre (table needed for mapping items and genres), Numbers (a helper table), Occupation, PredictedRatings, PredictedItems, PredictionMethod, TestRatings, TestItems, TrainRatings, TrainItems, and User. The Item and User table are in fact views over the master data provided with the Movielens dataset and dynamic information gathered from usage data. Further views are SquareError, UserwiseFMeasure, AllRatings, and AgeAnalysis.

On top of the warehouse prototype, an OLAP cube for rating prediction was created using Microsoft SQL Server Analysis Services. Within this cube, the respective measures were created: counts and sums, and further derived measures such as distinct counts, averages, standard deviations, ranks, (running) differences and (running) percentage. The core measures RMSE and MAE are derived from the error between predicted and actual ratings. The most important OLAP task with respect to framework development is to define the relationships between the measures and dimensions, as several dimensions are linked multiple times (e.g. the Age dimension is role-playing as it is linked against both item age and user age) or only indirect relationships exist (such as between category and rating the relationship is only established via item). Designing the relationships has to be exercised very carefully, as both correctness of the model and the ability to programmatically navigate dimensions and measures (adding them on the report axes, measure field or as filters) depend on this step. Linking members enables generic dimensions such as Prediction Method A, and Prediction Method B, that can be linked to chosen dimension members. This renders unnecessary the creation of the $n(n-1)/2$ possible measures yielding differences between any two prediction methods A and B (for, say, RMSE or F-measure). Furthermore, this approach allows choosing more than one dimension member, e.g. several runs of one algorithm with different parameters, as one linked member for aggregate analysis.

Before we go on to the evaluation of our prototype, let us state that our framework describes more than simply a model for designing evaluation frameworks. The prototype serves well as a template for other recommender datasets, too. With nothing changed besides the data load procedure, it can be used directly for, e.g., the other Movielens datasets, the Netflix challenge dataset or the Eachmovie dataset. Additional data available in those datasets (e.g. the tagging information from the Movielens 10m dataset) are either ignored or require an extension of the data warehouse and the multidimensional model (resulting in new analysis possibilities).

6. PERFORMANCE EVALUATION

In the previous section we have described the implementation of a research prototype of the proposed framework using the Movielens 1m dataset. Building on this prototype, we proceed with presenting a set of results that are obtained by applying it.

We have to clarify that the objective of our experimental evaluation is not limited to the comparison of specific recommender algorithms, as it is mostly performed in works that propose such algorithms. Our focus is, instead, on demonstrating the flexibility and easiness with which we can answer important questions for the performance of recommendations. It is generally agreed that explicitly modelling the effects describing changes in the rating behavior over the various users (user base-effect), items (item base-effect), and age of the respective item or user (time effects) [3, 16, 17]. For this reason, we choose to demonstrate the benefits of the proposed framework by setting our scope on those effects followed by exemplary dissecting the performance of two widely examined classes of recommender algorithms, i.e., collaborative filtering and matrix factorization. We also consider important the exploratory analysis of items and users, which can provide valuable insights for business analysts about factors determining the performance of their recommender systems. We believe that the results presented in the following demonstrate how easy it is to obtain them by using the proposed framework, which favors its usage in real-world applications, but also can provide valuable conclusions to motivate the usage of the framework for pure research purpose, since it allows for observing and analyzing the performance by combining all related dimensions that are being modeled.

All results presented in the remainder of this section could easily be obtained graphically by navigating the presented measures and dimensions using Excel 2007 as multidimensional client.

6.1 Exploratory Data Analysis

Using the framework, the first step for a research and a business analytics approach is exploring the data. As an example, the Calendar dimension (Date) is used to slice

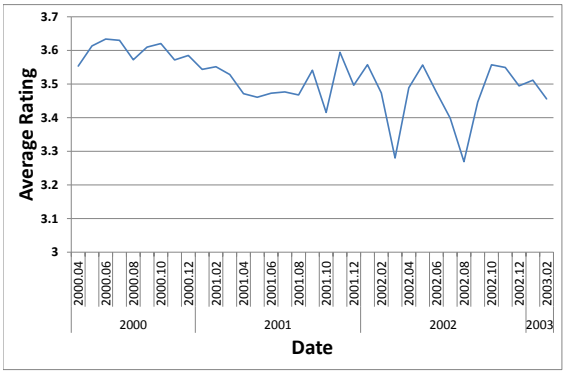


Figure 2: Average rating by date

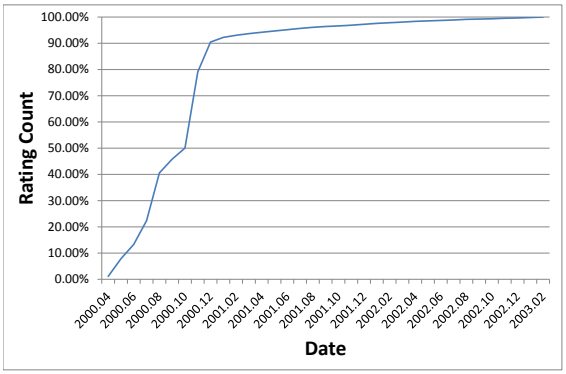


Figure 3: Rating count by date (running percentages)

the average rating measure. Figure 2 presents this as pivot chart. The sharp slumps noticeable in March and August 2002 together with a general lack of smoothness beyond mid 2001 arouse curiosity and suggest replacing average rating by rating count (figure not shown). Changing from counts to running percentages proves that about 50 percent of the ratings in this dataset are spent within the first six months out of nearly three years. Within two more months 90 percent of the ratings are assigned, roughly seven percent of the data for 50 percent of the time (figure 3).

6.1.1 Item Analysis

The framework allows an easy visualization of the item effect described e.g. in [16], namely that there usually is a systematic variation of the average rating per item. Aditionally, other factors can easily be integrated in such an analysis. Figure 4 shows the number of ratings received per item sorted by decreasing average rating. This underlines the need for regularization when using averages, as the movies rated highest only received a vanishing number of ratings.

Moving on the x-axis from single items to rating count buckets containing a roughly equal number of items, a trend of heavier rated items being rated higher can be observed (figure omitted for space reasons). A possible explanation might be that blockbuster movies accumulate a huge number of generally positive ratings during a short time and the all-time classics earn a slow but steady share of additional coverage. That all-time classics receive higher ratings can nicely be proved with the framework, too. Consistent with findings during the final phase of the Netflix competition by

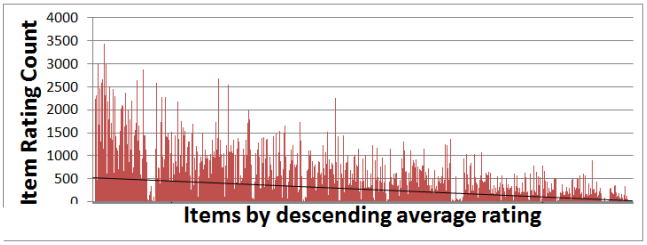


Figure 4: Item rating count sorted by decreasing average rating

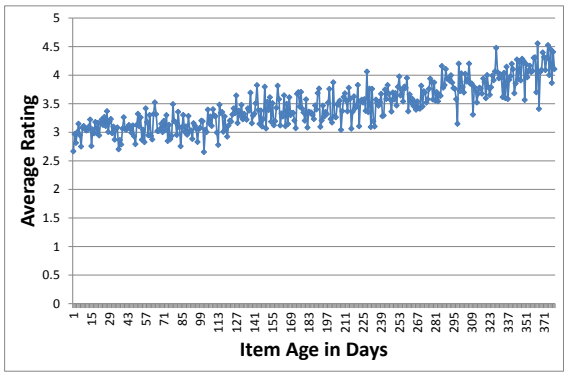


Figure 5: The all-time classics effect. Ratings tend to increase with the age of the movie at the time the rating is received. Age is measured in time since the first rating recorded.

Koren [17], figure 5 shows a justification for the good results obtained by adding time-variant base effects to recommender algorithms. Besides the all-time classics effect, the blockbuster effect can also be observed (figure 6), showing that items who receive numerous ratings per day on average also have a higher rating.

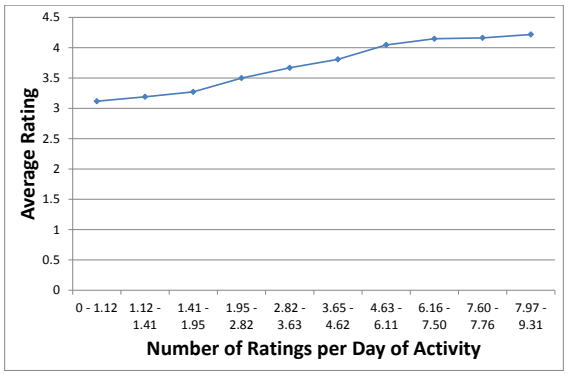


Figure 6: The blockbuster effect. Increasing average item rating with increasing number of ratings received per day.

Slicing the average rating by Genre shows a variation among the different genre with Film-Noir being rated best (average rating 4.07, 1.83% of ratings received), and Horror being rated worst (3.21, 7,64%). Of the Genres with at least ten percent of the ratings received Drama scores highest (3.76, 34.45%) and Sci-Fi lowest (3.46, 15.73%). Figure not shown.

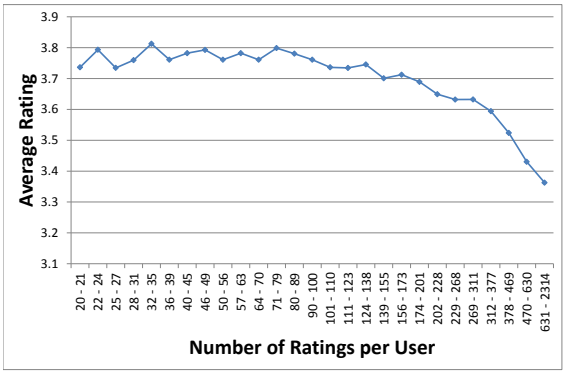


Figure 7: The effect of the number of ratings per user on the average rating

6.1.2 *User Analysis*

The user effect can be analyzed just as easy as the item effect. Reproducing the analysis explained above on the users, it is interesting to notice that for heavy raters the user rating count effect is inverse to the item rating count effect described above (figure 7): the higher the amount of ratings spent by a given user, the lower his or her average rating. One explanation to this behavior might be that real heavy raters encounter a lot of rather trashy or at least low quality movies.

6.2 Recommender Model Diagnostics

For algorithm performance comparison, the Movielens 1m ratings were randomly split into two nearly equal size partitions, one for training (500103), and one for testing (500104 ratings). Algorithm parameter estimation was conducted on the training samples only, predictions were conducted solely on the test partition. Exemplarily, a vanilla matrix factorization (20 features, regularization 0.09, learn rate 0.01, 56 iterations, hyperparameters optimized by 5-fold cross-validation) is analyzed.[2]

For a researcher the general aim will be to improve the overall RMSE or F-Measure, depending on the task, as this is usually what wins a challenge or raises the bar on a given dataset. For a business analyst this is not necessarily the case. A business user might be interested in breaking down the algorithm's RMSE over categories or top items or top users as this may be relevant information from a monetary aspect. The results of the respective queries may well lead to one algorithm being replaced by another on a certain part of the dataset (e.g. subset of the product hierarchy).

In figure 8, RMSE is plotted vs. item rating count in train.This indicates that more ratings on an item do help factor models. Interpreted the other way around, for a business user, this implies that this matrix factorization yields best performance on the items most crucial to him from a top sales point of view (though for slow seller other algorithms might be more helpful).

The same trend can be spotted when RMSE is analyzed by user rating count on the training set (figure omitted for space reasons), though the shape of the curve follows a straighter line than for the item train rating count (where it follows more an exponential decay).

Due to the approach taken in the design of the OLAP cube the number of recommender algorithms comparable as A and B is not limited; neither does it have to be exactly one algorithm being compared with exactly one other, as multiple selection is possible. Furthermore—given the predictions are already in the warehouse—replacing one method by another or grouping several methods as A or B can nicely be achieved by selecting them in the appropriate drop-down list. Exemplarily, the matrix factorization analyzed above is compared to the global average of ratings as baseline recommendation method. Figure 9 reveals that for this factor model more ratings on train do increase the relative performance, as expected, up to a point from which the static baseline method will gain back roughly half the lost ground. Investigation of this issue might be interesting for future recommender models.

All results presented could be obtained very fast: when judging the time needed to design query and report (chart)—which was on average seconds for construction of the query and making the chart look nice—, and when judging execution time—which was in the sub-second timeframe.

[2]The matrix factorization yielded an RMSE of 0.8831 given the presented train-test split.

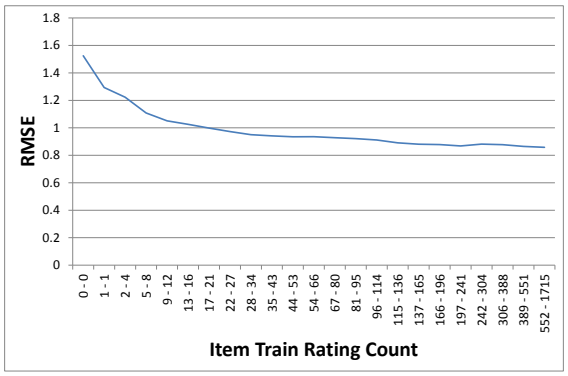


Figure 8: Item rating count effect on a factor model. Buckets created on roughly equal item count.

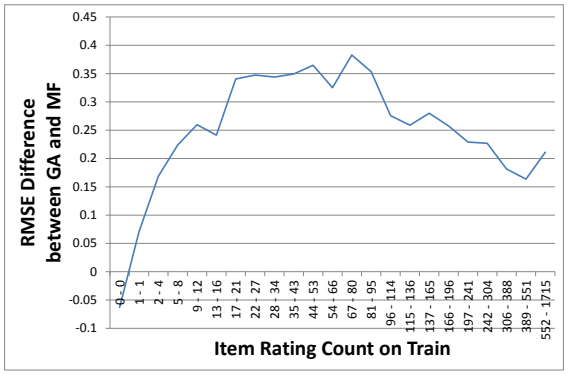


Figure 9: Difference in RMSE between Matrix Factorization (MF) and Global Average (GA) vs. ratings available per item on the train dataset.

7. CONCLUSIONS

We have proposed a novel multidimensional framework for integrating OLAP with the challenging task of evaluating recommender systems. We have presented the archi-

tecture of the framework as a template and described the implementation of a research prototype. Consistent with the other papers at this workshop, the authors of this work believe that the perceived value of a system largely depends on its user interface. Thus, this work provides an easy to use framework supporting visual analysis. Our evaluation demonstrates, too, some of the elegance of obtaining observations with the proposed framework. Besides showing the validity of findings during the recent Netflix prize on another dataset, we could provide new insights, too. With respect to the recommender performance evaluation and the validity of RMSE as an evaluation metric, it would be interesting to see if a significant difference in RMSE concerning the amount of ratings present in the training set would also lead to significant effects in a related user study.

In our future work, we will consider the extension of our research prototype and develop a web-based implementation that will promote its usage.

8. ACKNOWLEDGMENTS

The authors gratefully acknowledge the co-funding of their work through the European Commission FP7 project MyMedia (grant agreement no. 215006) and through the European Regional Development Fund project LEFOS (grant agreement no. 80028934).

9. REFERENCES

[1] G. Adomavicius, R. Sankaranarayanan, S. Sen, and A. Tuzhilin. Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Trans. Inf. Syst.*, 23(1):103–145, 2005.

[2] G. Adomavicius and A. Tuzhilin. Multidimensional recommender systems: A data warehousing approach. In *WELCOM '01: Proceedings of the Second International Workshop on Electronic Commerce*, pages 180–192, London, UK, 2001. Springer-Verlag.

[3] R. Bell, Y. Koren, and C. Volinsky. Modeling relationships at multiple scales to improve accuracy of large recommender systems. In *KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 95–104, New York, NY, USA, 2007. ACM.

[4] D. Bollen, B. P. Knijnenburg, M. C. Willemsen, and M. Graus. Understanding choice overload in recommender systems. In *RecSys '10: Proceedings of the 2010 ACM conference on Recommender systems*, New York, NY, USA, 2010. ACM.

[5] J. S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *MSR-TR-98-12*, pages 43–52. Morgan Kaufmann, 1998.

[6] O. Celma and P. Herrera. A new approach to evaluating novel recommendations. In *RecSys '08: Proceedings of the 2008 ACM conference on Recommender systems*, pages 179–186, New York, NY, USA, 2008. ACM.

[7] E. Codd, S. Codd, and C. Salley. Providing OLAP to user-analysts: An it mandate. Ann Arbor,MI, 1993.

[8] T. Crook, B. Frasca, R. Kohavi, and R. Longbotham. Seven pitfalls to avoid when running controlled experiments on the web. In *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1105–1114, New York, NY, USA, 2009. ACM.

[9] GroupLens. Movielens data sets. http://www.grouplens.org/node/73.

[10] C. Hayes, P. Massa, P. Avesani, and P. Cunningham. An on-line evaluation framework for recommender systems. In *In Workshop on Personalization and Recommendation in E-Commerce (Malaga*. Springer Verlag, 2002.

[11] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, 22(1):5–53, 2004.

[12] W. H. Inmon. *Building the Data Warehouse*. Wiley, 4th ed., 2005.

[13] T. Joachims, L. Granka, B. Pan, H. Hembrooke, and G. Gay. Accurately interpreting clickthrough data as implicit feedback. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 154–161, New York, NY, USA, 2005. ACM.

[14] N. Jones, P. Pu, and L. Chen. How users perceive and appraise personalized recommendations. In *UMAP '09: Proceedings of the 17th International Conference on User Modeling, Adaptation, and Personalization*, pages 461–466, Berlin, Heidelberg, 2009. Springer-Verlag.

[15] R. Kimball and M. Ross. *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling*. Wiley, 2nd ed., 2002.

[16] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *KDD '08: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 426–434, New York, NY, USA, 2008. ACM.

[17] Y. Koren. Collaborative filtering with temporal dynamics. In *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 447–456, New York, NY, USA, 2009. ACM.

[18] Microsoft. Microsoft SQL Server 2008 homepage. http://www.microsoft.com/sqlserver/2008/.

[19] J. O'Brien and G. Marakas. *Management Information Systems*. McGraw-Hill/Irwin, 9th ed., 2009.

[20] A. Thor and E. Rahm. Awesome: a data warehouse-based system for adaptive website recommendations. In *VLDB '04: Proceedings of the Thirtieth international conference on Very large data bases*, pages 384–395. VLDB Endowment, 2004.

[21] M. Zhang and N. Hurley. Avoiding monotony: improving the diversity of recommendation lists. In *RecSys '08: Proceedings of the 2008 ACM conference on Recommender systems*, pages 123–130, New York, NY, USA, 2008. ACM Press.

[22] C.-N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen. Improving recommendation lists through topic diversification. In *WWW '05: Proceedings of the 14th international conference on World Wide Web*, pages 22–32, New York, NY, USA, 2005. ACM.

An Evaluation of Multilabel Classification for the Automatic Annotation of Texts

Eneldo Loza Mencía
Knowledge Engineering Group
Technische Universität Darmstadt
eneldo@ke.tu-darmstadt.de

Abstract

This article presents the formulation of an information extraction as a multilabel classification problem. This representation allows for exploiting annotation overlappings and correlations. The standard multiclass approach is compared to different multilabel classification algorithms.

1 Introduction

In recent years, more and more approaches have appeared that translate the IE task into a classical classification problem, which is nowadays considered the most popular approach. The most common approach is to transform each text position, i.e. usually each text token in the document, into a classification example. This is often called boundary classification or sequential tagging/labeling. The class information of the instance depends on whether the underlying text token is a part or boundary of the target annotation or not. Figure 1 shows an example tagged sentence. The token *The* is marked as the beginning of a noun phrase ([NP) that ends at *fox* with NP]. The standard approach is to train one separate classifier for each annotation type, i.e. one for noun phrases, one for verb phrases etc. This subproblems can be solved using a multiclass classification algorithm that is trained to predict for each token exactly one class. We can often observe an overlapping of the annotations of the different types in real world applications, such as in chunking, syntactic parsing or ontology based information extraction (OBIE). The token *The* e.g. is at the same time a determiner and the beginning of a noun phrase, which is indeed linguistically a reasonable coincidence. The traditional approach ignores this co-occurrence and is therefore not able to exploit the additional information, namely that determiner are often also noun phrases beginnings.

The approach presented in this work therefore reformulates the many individual multiclass problems into only one multilabel classification problem. In contrast to multiclass, multilabel classification allows an instance to be associated to several classes, in this context often called labels. This means that a token is now allowed to have simultaneously several classes assigned. The purpose of this representation is twofold: on the one hand we obtain a more natural, compact and consistent representation of the extraction problem. On the other hand, the main purpose of this formulation is to allow an underlying multilabel algorithm to exploit relations in the labels such as co-occurrence, exclusions and implications and hence improve the prediction quality. This work evaluates several multilabel learning algorithms and compares them on a dense annotation dataset.

Only few attempts have been made on this subject so far. McDonald *et al.* [2005] were able to improve accuracy in extracting non-contiguous and overlapping segments of different types using an adapted multilabel algorithm. However, their algorithm is not directly comparable since it is centered and adapted to sentences whereas the approach presented here is token based and usable with any multilabel algorithm.

2 Preliminaries

The transformation of an information extraction task into a classification problem was already sketched in Section 1. The next two paragraphs give a more detailed description of the two main processing steps (mainly based on [Loza Mencía, 2009]), continued by the introduction of the employed multilabel classification algorithms (a recent overview can be found in [Tsoumakas *et al.*, 2009]).

Boundary classification In this work we employ the simple but effective *Begin/End* approach. The start and the end of each annotation, i.e. only the boundaries, are marked with the tag *BEGIN* or *END*, the rest is marked as negative examples with *NEG*. The bottom two rows in Figure 1 shows for each type DT (determiner), JJ (adjective), NN (noun) etc. the beginning and ending of an annotation.[1]

In the standard approach, a problem appears when an annotation only includes one token, such as for DT. This would make it necessary to tag a token as *BEGIN* and *END* simultaneously. As this would require a multilabel capable underlying classifier, the common approach is to include an additional class *UNIQUE* which represents both classes at the same time (see also Section 3).

We chose a pragmatic way for solving inconsistencies during the reconstruction of the annotations on the test set: we search for the first appearing of an opening tag and continue the extraction until the first matching closing tag is found, the remaining tags are ignored.

Feature Generation The boundary classification step generates the class information for each training instance, but up to now these instances are empty. The simplest features which can be added are the occurrences of the different tokens themselves. Since the focus of this work lies on the comparison of the classification algorithms, we use only these simple features and refrain from using sophisticated linguistic features. For the same reason we ignore the classification history.

[1] The negative class is omitted since this is the multilabel representation.

token	The	quick	brown	fox	jumps	over	the	lazy	dog
token	the=1	quick=1	brown=1	fox=1	jumps=1	over=1	the=1	lazy=1	dog=1
features	+1.quick=1	+1.brown=1	+1.fox=1	+1.jumps=1	+1.over=1	+1.the=1	+1.lazy=1	+1.dog=1	
		-1.the=1	-1.quick=1	-1.brown=1	-1.fox=1	-1.jumps=1	-1.over=1	-1.the=1	-1.lazy=1
POS	[DT, DT]	[JJ, JJ]	[JJ, JJ]	[NN, NN]	[VBZ, VBZ]	[IN, IN]	[DT, DT]	[JJ, JJ]	[NN, NN]
syntactic	[NP			NP]	[VP	[PP	[NP		NP], PP], VP]

Figure 1: Transformation of a text sentence into a classification problem. Each column shows a token and exemplarily the generated features with a context of one word and the class information of the corresponding generated classification instance. The first row of the class information 'POS' shows the part-of-speech annotations, the second the syntactic annotations given to the token. Abbreviations according to [Marcus *et al.*, 1993]. A '[' denotes the *BEGIN* and ']' the *END* of the corresponding annotation type.

The token features row in Figure 1 shows the type of windowing we used.

2.1 Multilabel Classification

We represent an instance or text position as a vector $\bar{\mathrm{x}} = (x_1, \ldots, x_M)$ in a feature space $\mathcal{X} \subseteq \{0,1\}^N$. In multilabel problems, each instance $\bar{\mathrm{x}}_i$ is assigned to a set of relevant labels $\mathcal{Y}_i$, a subset of the K possible classes $\mathcal{Y} = \{c_1, \ldots, c_K\}$, in contrast to multiclass problems, where each instance is mapped to exactly one class, i.e. $\|\mathcal{Y}_i\| = 1$.

Binary Relevance In the binary relevance (BR) or one-against-all (OAA) method, a multilabel problem with K possible classes is decomposed into K binary problems. For each subproblem, a binary classifier is trained to predict the relevance of the corresponding class.

QWeighted Calibrated Label Ranking QCLR is a recently proposed algorithm, which is an efficient approach for multilabel classification. This algorithms combines three aspects: the pairwise decomposition of multilabel problems, calibrated label ranking for determining multilabel result and an adaption of the QWEIGHTED algorithm for efficient prediction Loza Mencía *et al.* [2009]. In the pairwise decomposition approach, one classifier is trained for each pair of classes, i.e., a problem with K different classes is decomposed into $\frac{K(K-1)}{2}$ smaller subproblems. An example is added to the training set $c_u vs. c_v$ if c_u is a relevant class and c_v is an irrelevant class or vice versa, i.e., $(c_u, c_v) \in \mathcal{Y} \times \overline{\mathcal{Y}} \cup \overline{\mathcal{Y}} \times \mathcal{Y}$ with $\overline{\mathcal{Y}} = \mathcal{Y} \backslash \mathcal{Y}$ as negative label set. During classification, each base classifier is queried and the prediction is interpreted as a vote for one of its two classes. The resulting ranking of classes is split into relevant and irrelevant classes via calibration. The QWEIGHTED approach allows to reduce the classification costs from quadratic to log-linear time.

Label Powerset In the label powerset approach (LP), a meta multiclass problem is constructed where each appearing label combination $\mathcal{Y}_i$ is interpreted as one separate class. The meta problem is then solved with a normal multiclass algorithm or with the previously presented decomposition methods. In the worse case, the resulting multiclass problem has an increased amount of classes of $min(M, 2^K)$ where M is the number of training examples, however the number tends to be much smaller due to class correlations.

3 Multilabel Classification for Information Extraction

As already outlined, one of the advantages of multilabel classification is the more natural representation since we do not have to work with tricks. Note e.g. that in the *BEGIN/END/UNIQUE* scheme the learning algorithm is forced to learn to distinguish between *UNIQUE* and *BEGIN* resp. *END* though *UNIQUE* is actually a subset of these two classes. This makes this tagging scheme especially interesting for the usage of multilabel classification. Furthermore, the traditional methods do not permit to exploit relations between several annotation types since each type is by design necessarily learned separately. We present therefore in the following the standard approach together with the multilabel alternatives.

Traditional multiclass approach A multiclass classifier is trained for each annotation type, the *BEGIN/END/UNIQUE/NEG* scheme is used. I.e. for each annotation type $a \in A$ a classifier is trained with instances mapped to exactly one class c in $\mathcal{Y}_a = \{$*BEGIN, END, UNIQUE, NEG*$\}$. The multiclass problem is solved via one-against-all decomposition in our case.

Binary Relevance and QCLR The extraction task is transformed into one multilabel problem where each token is assigned to a subset $\mathcal{Y}$ of $\mathcal{Y} = A \times \{$*BEGIN, END*$\}$, with A as the set of annotation types. Note that it is not necessary to include the *NEG* as the algorithm is able to predict the empty set.

In the binary relevance setting, the algorithm is not expected to improve from label co-occurrences since each base classifier is trained separately. However, the pairwise approach is at least potentially able to detect non co-occurrences, since we train for each pair of classes a base classifier with instances where the one class is positive and the other negative, i.e. the base classifier is trained with cases where both classes are mutually exclusive. Therefore this approach may be able detect that a co-prediction of two classes is wrong for a determined instance, in contrast to BR, where a base classifier cannot state anything else than relevant or non-relevant. Recently, promising advances were made in enhancing the pairwise approach by the detection and exploitation of present constraints on the labels, such as co-occurrences [Park and Fürnkranz, 2008]. We are currently working on incorporating these ideas.

Label Powerset The multilabel problem is retransformed into a multiclass problem, i.e. the possible classes of a token are in $\mathcal{Y} = 2^{A \times \{BEGIN, END\}}$. Note that for only one annotation type this corresponds to the

traditional multiclass approach, since we would obtain $c \in \mathcal{Y} = \{\{\}, \{BEGIN\}, \{END\}, \{BEGIN,END\}\}$, which corresponds to $\{BEGIN, END, UNIQUE, NEG\}$. But for more than one annotation type, co-occurrence and implications can effectively be exploited and detected since these co-occurrences are explicitly used as training information. However, the granularity of this information is limited, since the approach is only able to abstract from the co-occurrence of two classes if there is no other class appearing since this would not generate the desired meta co-occurrence class anymore.

4 Evaluation

Since it is difficult to obtain densely annotated (free) corpora e.g. from the field of OBIE, we decided to generate our own dense dataset with the help of the Stanford Parser, which returns the syntactic structure of a sentence.[2] The result of the parser was considered to be the true and correct labeling of the corpus. The first six (scientific) texts from the *Learned* category of the Brown Corpus [Francis and Kucera, 1979] were annotated with this tool, taking the first three documents for training and the remaining for testing. The resulting training set contains 6790 instances and 48 different annotations types, 7091 instances remained for testing. Since each annotation type leads to two tags denoting the *BEGIN* and the *END*, we obtain 96 different labels for the multilabel problem. In average there are 3.34 labels associated per token. For the label powerset representation, 334 classes were retrieved. A window size of 5 resulted in less than 7000 different features. We used the well known LibSVM library with linear kernel and standard settings as our base learner [Chang and Lin, 2001].

The results are shown in Table 4. The first observation is that the multilabel approaches (QCLR and BR) seem to slightly outperform the traditional multiclass approach in terms of F1, and that ML has a slight advantage over LP. A closer look reveals that recall and precision highly depend on the used transformation approaches. LP seems to boost recall while ML and especially the classical multiclass approach improve precision, always at the expense of the opposite measure. The MC setting appears to generate quite conservative classifiers, since the MC extractor predicts 18% less annotations than ML-QCLR and even 28% less than LP-PC.

Note that the absolute values may seem generally low, but remind that these results were produced without linguistic or any other intelligent preprocessing. Moreover, only exact annotation matches were taken into account, counting token matches improves the rates to around 70%.

5 Conclusions

We have presented the approach of presenting an information extraction problem as one multilabel classification problem rather than several independent multiclass problems. This view is more natural to the extraction problem and furthermore potentially allows the exploitation of relations and correlations between overlapping annotations.

Although all multilabel approaches achieve higher F1 scores in the experiments than the standard approach, a direct comparison of both approaches shows up to be difficult, since the traditional approach is focused on precision while the multilabel approaches obtained higher recall to the extend of the precision. Evaluations on more corpora and perhaps with a more extensive also linguistic preprocessing are planned in order to obtain a clearer picture. Nevertheless, it has been demonstrated that the formulation as multilabel problem is at least comparable, particularly considering that the employed algorithms are not especially adapted or designed in order to exploit class correlations. Recent advantages in the relatively new field of multilabel classification let us expect substantial improvements (cf. [Tsoumakas *et al.*, 2009]). Furthermore, we are currently investigating an adaption of the pairwise approach that benefits from restrictions and constraints on the possible class constellations [Park and Fürnkranz, 2008].

Algorihm	Precision	Recall	F1
MC	74.21	34.32	46.93
ML-BR	72.49	40.52	51.98
ML-QCLR	71.49	40.18	51.44
LP-BR	59.67	43.46	50.29
LP-PC	65.12	41.73	50.87

Table 1: Prediction quality of the different algorithms based on exact annotation matches, micro-averaged over all annotation types. MC for the traditional multiclass algorithm, ML for the multilabel transformation and LP for label powerset. PC denotes the pairwise decomposition approach for multiclass problems.

6 References

Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

W. N. Francis and H. Kucera. Brown corpus manual. Technical report, Department of Linguistics, Brown University, Providence, Rhode Island, US, 1979.

Eneldo Loza Mencía, Sang-Hyeun Park, and Johannes Fürnkranz. Efficient voting prediction for pairwise multilabel classification. In *Proceedings of the 11th European Symposium on Artificial Neural Networks (ESANN-09)*, 2009.

Eneldo Loza Mencía. Segmentation of legal documents. In *Proceedings of the 12th International Conference on Artificial Intelligence and Law*, pages 88–97, 2009.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, 19(2):313–330, 1993.

Ryan T. McDonald, Koby Crammer, and Fernando Pereira. Flexible text segmentation with structured multilabel classification. In *Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, 2005.

Sang-Hyeun Park and Johannes Fürnkranz. Multi-label classification with label constraints. In *Proceedings of the ECML/PKDD-08 Workshop on Preference Learning (PL-08)*, pages 157–171, 2008.

Grigorios Tsoumakas, Ioannis Katakis, and Ioannis Vlahavas. Mining multi-label data. In *Data Mining and Knowledge Discovery Handbook*. Springer, 2009.

[2] http://www-nlp.stanford.edu/software/lex-parser.shtml

Pattern Mining in Sparse Temporal Domains, an Interpolation Approach Work in Progress

Christian Pölitz
University of Bonn
poelitz@iai.uni-bonn.de

Abstract

Weblog systems, mobile phone companies or GPS devices collect large amounts of personalized data including temporal, positional and textual information. Patterns extracted from such data can give insight in the behavior and mood of people. These patterns are often imprecise due to sparseness in the data. We propose an interpolation technique that augments local patterns with elements that seem to be locally unimportant but with global information they are interesting.

1 Introduction

Large amounts of data are gathered in social web applications, on mobile phone calls, GPS signals from cars or animals, to name only a few. The companies that possess these data are highly interested in benefiting from the contained information. One way to extract such information is to find patterns. By patterns we mean regularities like frequently mentioned topics in weblogs, paths that certain animals take for foraging or main traffic routes.

All the data that we investigate include temporal information like the time a weblog is written, the time stamp of a telephone call or the time of GPS signals. This leads to two characteristics for the patterns we are interested in. First the patterns must be temporally ordered, for instance which topics follow a certain frequent topic in weblogs or frequently consecutively visited places. These patterns are called sequential patterns and several algorithmical approaches exist to solve this task. An overview of different sequential pattern mining methods and some special cases are given by Zhao and Bhowmick in [Q. Zhao, 2003].

The second characteristic is that the patterns have a temporal extend and transition times. This means each pattern has a clear beginning time and an end time. All contained subpatterns have also a beginning and an end. From this we indirectly know the time interval of the pattern. The transition time is the time interval between two consecutive elements of the pattern and can be calculated from the temporal extend of the subpatterns.

Possible patterns can be the evolution of topics in blog entries. Such information is useful for many different companies. They can directly search for such topic patterns that deal with their products or services. Other patterns could be movements of groups of people in certain areas. These patterns can be used for targeted advertisements by poster campaigns. To know where certain people move could influence the kind of commercial being shown on posters.

One big problem by extracting patterns from the data is sparseness. Sparseness means that the data contains large temporal gaps and/or very few data points with an assignment to one specific identifier like blog entry author, mobile phone device or GPS device. Such weaknesses in the data can badly influence the search for local patterns. As a result we may get patterns with very large temporal intervals with no information of the intermediate time. We try to deal with this problem by using global information to interpolate between consecutive pattern elements with a big temporal difference. The global information can be of arbitrary source, additionally or data immanent.

id	topics		
1	A	B	C
2	A		C
3	A	B	
4		B	C
5	A		C
6	A	B	C
	$[s_1, e_1]$	$[s_2, e_2]$	$[s_3, e_3]$
	time intervals		

Table 1: A possible development of topics in a weblog.

In weblogs for instance, "bloggers" might write entries quite rarely or there are large breaks between entries. This can for example be due to individual behavior or holidays. For these times it might be impossible to retrieve local patterns. We try to overcome this problem by using global information of all bloggers in the corresponding times. We simply use the assumption that the hot topics in the time in which we lack local information would also track the attention of the corresponding bloggers.

On table 1 we see a possible distribution of topics A, B and C that are discussed by six different bloggers in certain time intervals. We see that two of three persons who write about topic A in time interval $[s_1, e_1]$ do write later on in time interval $[s_3, e_3]$ about topic C. This leads to the rule if someone writes about topic A, they also write about topic C in the given times with a confidence of four fifth.

For the intermediate time between the end of topic A e_1 and the beginning of topic C s_3 there is topic B. Only two persons write about topic B in the mean time after writing about topic A and before topic C. The data supports this only by one third. Applying strictly a pattern mining method we could loose this intermediate topic due to the low support.

From the data itself there is the possible pattern of writing about topic A in time interval $[s_1, e_1]$ leads to writing about topic B in $[s_2, e_2]$. Further there is the pattern of people writing about topic B in $[s_2, e_2]$ will later write

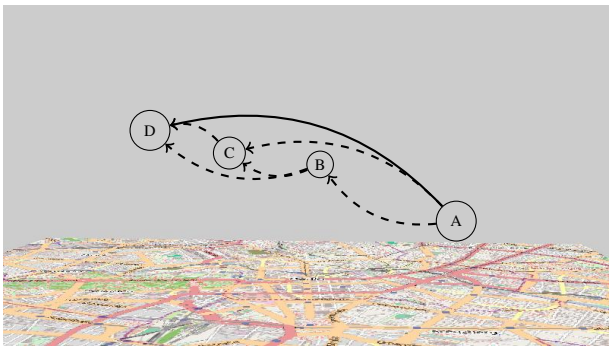


Figure 1: Possible movement pattern in geographical space and in time.

about topic C in $[s_3, e_3]$. Both patterns have a confidence of three fifth. Since topic B was mentioned four times in the time between A and C we augment the previous pattern by: if people write about topic A and C in the corresponding times they will also write about topic B in the intermediate time $[s_2, e_2]$.

From this example we gain the following knowledge. A possible interpolation of a pattern (or augmentation) should only contain elements (topics) with at least a certain amount of appearance. Additionally the interpolated topics should by at least mentioned by some of the persons that form the previous (original) pattern.

An other example that concentrates on movement patterns of GPS data is shown on figure 1. Four places are visited in different times indicated by the height of the nodes. The GPS signals have unequal dwell times which is shown by different sizes of the nodes. The solid directed line shows a possible pattern among the places extracted from the GPS data. This pattern leads to the rule that if someone was at place A at the specific time they will be later at place D at another specific time.

Due to unregularities and annoyances there is a large temporal and spatial gap between place A and D. From the whole data set we know that in the intermediate time the places C and B are frequently visited and some of the people that support the previous patterns have been at this times at these places, but not enough to make this pattern interesting. From this information we augment the pattern by the places C and B. The resulting pattern is shown on figure 1 as dashed directed lines.

2 Pattern Extraction Method

The task to solve the above stated problems is a clear pattern mining task. An introduction in pattern mining and a survey on several algorithms is described by Goethals in [Goethals, 2003].

First we describe definitions about the patterns we want to find. We include constrains that make sure we retrieve reasonable patterns from data from a temporal domain containing additional information like texts from blog entries or GPS coordinates of cars.

Generally we define a pattern as $P = i_1[s_1, e_1], i_2[s_2, e_2], \cdots, i_n[s_n, e_n]$. i_j are the basis elements of the patterns, analogue to the items in frequent item set mining. The elements are for instance topics in weblogs, mobile radio cells or clusters of GPS signals. Each element has a starting time s_j and an ending time e_j. This means at this time interval the topics are frequently used in weblogs, many phone calls were made in radius of a cell or lots of GPS signals are received. Analogue to sequential pattern mining we assume the elements to be temporally ordered, i.e. $s_k \leq s_l$ for $k < l$.

We define the elements i_j as disjoint subsets of the analyzed data set $D = \{o_1, \cdots, o_n\}$. Each subset contains only data points with a similarity larger than a given threshold. The similarity must be defined w.r.t. to the domain of the data and the application. For instance, if the data are blog entries and we are interested in patterns of topics, the similarity could be the semantical alikeness of the words in the entries to predefined or learned topics. In this case each subset would contain all the blog entries that contain words that indicate a certain topic.

In case the data are mobile phone calls, the similarity should be the mobile radio cell the phone is connected to (We assume that the mobile phones are only connected to one cell at a time). This means all mobile phone calls in a certain cell are highly similar and are contained in the same subset.

For GPS data of cars the similarity could be the geographical distance to certain streets or to concentrations of many other cars (clusters). A grid (even or uneven) division of the traveled area is also possible. Hence all GPS data of cars within a specific grid cell are similar.

These are only a few possible implementations of a similarity. In general we define the elements as described in equation 1 with a similarity measure $sim : D \times D \rightarrow \Re$ and a user defined (domain and application dependent) threshold parameter τ.

$$i_j = \{o_{1_j}, \cdots, o_{n_j} | o_{i_j} \in D, sim(o_{k_j}, o_{l_j}) \geq \tau\} \quad (1)$$

Among the elements, patterns are extracted w.r.t. a frequency measure. The measure depends on the uniqueness of the data points. For that we assume the data points to be assigned to an identifier id_j. Possible identifiers are the persons that wrote blog entries, mobile phones or GPS devices of cars. We generally assume the following structure of data points $o_j = (X, t, id)$ with X a vector information, t a time stamp and id the identifier. The information X can be the textual content of a blog entry, the cell information of a mobile radio cell or the position of a GPS device at time t.

A one element pattern $i_j[s_j, e_j]$ is frequent when it contains more than a predefined threshold n_{min} many unique identifiers. Additionally all contained data points have a time stamp $t \in [s_j, e_j]$ and are similar w.r.t. the above mentioned similarity measure sim.

A continuation of an $n-1$ elements pattern $P = i_1[s_1, e_1], \cdots, i_{n-1}[s_{n-1}, e_{n-1}]$ are the elements $i_{n_j}[s_{n_j}, e_{n_j}]$ with the following properties:

- $i_{n_j}[s_{n_j}, e_{n_j}]$ contains at least n_{min} data points with identifiers that are also contained in $i_{n-1}[s_{n-1}, e_{n-1}]$
- all data points in $i_{n_j}[s_{n_j}, e_{n_j}]$ with identifier id_j are temporally after the data point in $i_{n-1}[s_{n-1}, e_{n-1}]$ that have the identifier id_j too

Since the time span of the elements can be very large we partition the temporal information in intervals. These intervals can be periods with many blog entries to a certain topic, times with lots of phone calls or congestion of cars. A calendrical partitioning for instance in days or weeks is also possible.

To find such partitions we define an additional temporal similarity on the time information of the data points. All data points $o_k \in i_j$ must have a similarity $sim_t : D \times$

$D \to \Re$ larger than a predefined threshold τ_t (see equation 2). Together with the previous statements we define the elements i_j in equation 2.

$$i_j = \{o_{1_j}, \cdots, o_{n_j} | o_{i_j} \in D, sim(o_{k_j}, o_{l_j}) \geq \tau, \\ sim_t(o_{k_j}, o_{l_j}) \geq \tau_t\} \quad (2)$$

We include this statement in our pattern definition. For the one element pattern $i_j[s_j, e_j]$ we have the additional restriction that the contained data points are similar w.r.t. the temporal similarity measure sim_t. The continuations $i_{n_j}[s_{n_j}, e_{n_j}]$ have the additional property:

- all data points in $i_{n_j}[s_{n_j}, e_{n_j}]$ with identifier id_j that are in $i_{n-1}[s_{n-1}, e_{n-1}]$ too must be similar w.r.t. to the temporal similarity sim_t

The made definitions can be easily integrated into a sequential pattern mining method like PrefixSpan. In [Pei *et al.*, 2001] Pei et al. introduce the sequential pattern mining algorithm PrefixSpan that generates sequential patterns by growing prefixes of patterns. Such a method extracts all sequential patterns that are supported by at least n_{min} identifiers. For blog entries this means that at least n_{min} different persons must have written articles to the topics in the found patterns in the specified times. In case of GPS or telephone data the places in the patterns must be visited in the corresponding time intervals by at least n_{min} different GPS or telephone devices.

3 Pattern Extraction of Sparse Data

In the case of sparse data the pattern extraction becomes more difficult as we described above. In this case sparseness means that only very few data points share the same identifier while on the other hand there are many different identifiers. The data points with the same identifiers additionally have large differences in their temporal attributes.

To cope with these shortcomings we suggest an interpolation technique that uses all data points to find descriptions of the time between data points with the same identifier. According to a pattern extraction method we infer intermediate sequence elements of the patterns. This is done by allowing to add to a pattern sequence $P = i_1[s_1, e_1], \cdots, i_{n-1}[s_{n-1}, e_{n-1}]$ not only elements having enough data points with identifiers also contained in element $i_{n-1}[s_{n-1}, e_{n-1}]$ but having enough other data points with a very large similarities.

For possible continuations $i_{n_j}[s_{n_j}, e_{n_j}]$ of a pattern $P = i_1[s_1, e_1], \cdots, i_{n-1}[s_{n-1}, e_{n-1}]$ we state that in case the temporal difference $|e_{n-1} - s_{n_j}|$ exceeds a predefined threshold Δt and there exists no other $i_{n_{j'}}[s_{n_{j'}}, e_{n_{j'}}]$ that is temporally between e_{n-1} and s_{n_j} with the properties for a continuation, we insert between $i_{n-1}[s_{n-1}, e_{n-1}]$ and $i_{n_j}[s_{n_j}, e_{n_j}]$ interpolated subpatterns $P^* = i_1^*[s_1^*, e_1^*], \cdots, i_n^*[s_n^*, e_n^*]$ that result in an augmentation of the pattern to: $i_{n-1}[s_{n-1}, e_{n-1}], P^*, i_{n_j}[s_{n_j}, e_{n_j}]$.

By Δt we try to generate reasonable patterns with no more than this threshold of time between consecutive pattern elements. Depending on how much the temporal difference of two such pattern elements extends the threshold we allow to augements the pattern by less supported elements. For that we introduce a new parameter n_{min}^* (see **??**.

$$n_{min}^* = \frac{\Delta t \cdot n_{min}}{s_{n-1} - e_{n_j}} \quad (3)$$

The first element $i_1^*[s_1^*, e_1^*]$ of the pattern P^* contains more than a predefined threshold n_{min} many unique identifiers of which at lest n_{min}^* many are also in $i_{n_j}[s_{n_j}, e_{n_j}]$. Additionally all contained data points are temporally between e_{n-1} and s_{n_j}, further they are similar w.r.t. the above mentioned similarity measures sim and sim_t.

The continuation of an $n-1$ elements subpattern $P^* = i_1^*[s_1^*, e_1^*], \cdots, i_{n-1}^*[s_{n-1}^*, e_{n-1}^*]$ are the elements $i_{n_j}^*[s_{n_j}^*, e_{n_j}^*]$ with the following properties:

- $i_{n_j}^*[s_{n_j}^*, e_{n_j}^*]$ contains at least $n_{min}^* < n_{min}$ data points with identifiers that are also contained in $i_{n-1}[s_{n-1}, e_{n-1}]$ and in $i_{n_j}[s_{n_j}, e_{n_j}]$
- all together $i_{n_j}^*[s_{n_j}^*, e_{n_j}^*]$ contains at least n_{min} data points with different identifiers
- all data points in $i_{n_j}^*[s_{n_j}^*, e_{n_j}^*]$ with identifier id_j are temporally after the data points in $i_{n-1}[s_{n-1}, e_{n-1}]$ that have the identifier id_j too and later than the ones in $i_{n_j}[s_{n_j}, e_{n_j}]$
- all data points in $i_{n_j}^*[s_{n_j}^*, e_{n_j}^*]$ must be similar w.r.t. the similarity measure sim
- all data points in $i_{n_j}[s_{n_j}, e_{n_j}]$ with identifier id_j that are in $i_{n-1}[s_{n-1}, e_{n-1}]$ too must be similar w.r.t. to the temporal similarity sim_t

These definitions can be easily integrated in our previously stated pattern extraction algorithm. For a continuation of a pattern with large temporal difference as described above we simply apply again a sequential pattern mining method like PrefixSpan, but only to data points that have timestamps in the corresponding time interval.

A schematic example of the interpolation and augmentation of patterns of weblogs is shown in table 2 and for GPS movement patterns on figure 2.

As quality for the interpolated elements we use a similarity measure sim_{inter} based on the differences $d(i_j^*, i_{j+1}^*)$ between consecutive elements i_j^*, i_{j+1}^*. We generally assume that such elements do not differ too much. This assumption is similar to many interpolation techniques that require smooth interpolating functions. For movement data the difference could be the geographical distance and for weblog semantical difference between topics. Finally the similarity of the elements is the corresponding difference divided by their temporal difference $s_{j+1}^* -, e_j^*$ (see 4).

$$sim_{inter}(i_j^*, i_{j+1}^*) = \frac{d(i_j^*, i_{j+1}^*)}{s_{j+1}^* - e_j^*} \quad (4)$$

4 Future and Ongoing Work

Currently we are applying our proposed interpolation method on several data sets. Experiments on GPS signals show promising results. In this case it is easier to interpret the results since they can be shown on maps and the patterns are accustomed movements. The results on a large data set of GPS signals from cars are shown on figure 3. The patterns are extracted among concentrations of cars in Milan in the morning.

There is a pattern of 5 cars starting at place A from 4:06 am to 6:52 am reaching place E from 6:10 am to 7:09 am. The corresponding data points that form this pattern have a large temporal and a large spatial difference. W.r.t. the found patterns it seems very likely that cars moving from A to E went there by B, C and/or D. Applying our method for

pattern	hot topics among weblog entries							
P	$\cdots$	i_{n-1}						i_n
P_1^*			i_1^*	i_2^*	$\cdots$	i_{n-1}^*	i_n^*	
P_2^*			i_1^*	i_2^*	$\cdots$	i_{n-1}^*		
P_3^*			i_1^*	i_2^*	$\cdots$			
				$\cdots$				
$P * P_1^*$	$\cdots$	i_{n-1}	i_1^*	i_2^*	$\cdots$	i_{n-1}^*	i_n^*	i_n
$P * P_2^*$	$\cdots$	i_{n-1}	i_1^*	i_2^*	$\cdots$	i_{n-1}^*		i_n
$P * P_3^*$	$\cdots$	i_{n-1}	i_1^*	i_2^*	$\cdots$			i_n
				$\cdots$				
		$[s_{n-1}, e_{n-1}]$	$[s_1^*, e_1^*]$	$[s_2^*, e_2^*]$	$\cdots$	$[s_{n-1}^*, e_{n-1}^*]$	$[s_n^*, e_n^*]$	$[s_n, e_n]$
	time intervals							

Table 2: A schematic representation of interpolated patterns of topics in weblogs.

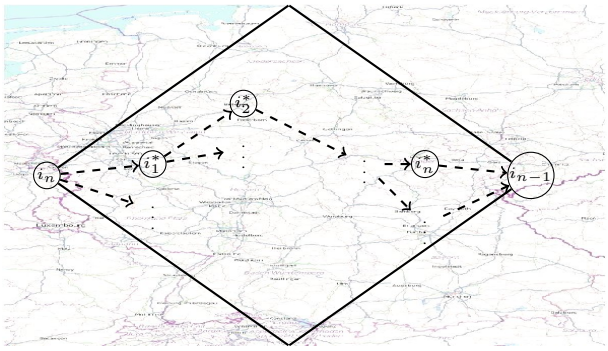


Figure 2: A schematic representation of a possible interpolation of two consecutive pattern elements with large temporal/spatial distance.

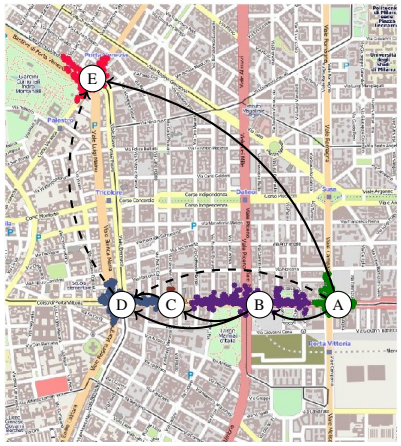


Figure 3: Augemented patterns among concentrations of cars.

pattern extraction on sparse data with $n_{min}^* = 2$ we retrieve additional patterns as shown by the dashed lines on figure 3. There are 2 of the 5 cars from the original pattern that went from A to D to eventually arrive at E. Beside these 2 cars there were several others cars in the intermediate time from 5:05 am to 5:54 am. We can now augment the pattern of moving from A to E by place D.

On weblog data an interpolation is harder to validated. We plan to artificially include sparseness in data. We want to analyze how good the augmented patterns are compared to corresponding patterns found without artificial sparseness. To achieve this we use retrieved patterns from weblog that have no sparse elements w.r.t. the statements above. Furthermore we insert sparseness into the data on data points that support the previously found patterns. On different degreess of sparseness we want to analyze how good the augmented patterns can reconstruct the original ones and how much they differ.

In conjunction with that, we plan to further investigate the influence of the parameters n_{min} and n_{min}^*. We hope to find heuristic descriptions of possible settings of these parameters concerning the domain of the data, additional (statistical) information and the application.

An other important aspect is the computation time of the pattern extraction and the interpolation. We generally assume the data to be ordered in time. By this the patterns can be found in a faster way and less main memory is used since we must only consider data with timestamps larger than the last element of the current pattern. Although this means in worst case we have to scan the whole data ($O(n)$) generally we assume local patterns to be small enough and short in time to be easily placed in main memory. For the interpolation we even expect subpattern with very few elements. That is due to the fact that only a smaller number of data points with timestamps between the corresponding elements of the original pattern might exists compared to the whole data set. Different data structures and data sources will be investigated in this context.

The propose method seems very promising to find patterns among large amounts of data from a temporal domain with additional sparseness. We are planning to show this on many different data sets in future works.

5 Aknowledgement

The work has been done within the research project ViAMoD Visual Spatiotemporal Pattern Analysis of Movement and Event Data, which is funded by DFG Deutsche Forschungsgemeinschaft (German Research Foundation) within the Priority Research Programme Scalable Visual Analytics (SPP 1335). We also thank the anonymous reviewers for their helpful suggestions.

References

[Goethals, 2003] B. Goethals. Survey on frequent pattern mining. Technical report, 2003.

[Pei *et al.*, 2001] Jian Pei, Jiawei Han, Behzad Mortazavi-asl, Helen Pinto, Qiming Chen, Umeshwar Dayal, and Mei chun Hsu. Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth. pages 215–224, 2001.

[Q. Zhao, 2003] S.S. Bhowmick Q. Zhao. Sequential pattern mining: a survey. Technical report, Technical Report Center for Advanced Information Systems, School of Computer Engineering, Nanyang Technological University, Singapore, 2003.

SVM Classifier Estimation from Group Probabilities

Stefan Rüping
Fraunhofer IAIS
Schloss Birlinghoven, 53754 St. Augustin, Germany
stefan.rueping@iais.fraunhofer.de

Abstract

A learning problem that has only recently gained attention in the machine learning community is that of learning a classifier from group probabilities. It is a learning task that lies somewhere between the well-known tasks of supervised and unsupervised learning, in the sense that for a set of observations we do not know the labels, but for some groups of observations, the frequency distribution of the label is known. This learning problem has important practical applications, for example in privacy-preserving data mining. This paper presents an approach to learn a classifier from group probabilities based on support vector regression and the idea of inverting a classifier calibration process. A detailed analysis will show that this new approach outperforms existing approaches[1]

1 Introduction

A learning problem that has only recently gained attention in the machine learning community is that of learning a classifier from group probabilities [Kueck and de Freitas, 2005; Quadrianto *et al.*, 2008; 2009]. It is a learning task that lies somewhere between the well-known tasks of supervised and unsupervised learning, in the sense that for a set of observations we do not know the labels, but for some groups of observations, the frequency distribution of the label in the groups is known (see Figure 1). The goal is, from this information alone, to estimate a classifier that works well on the labeled data.

As noted in [Quadrianto *et al.*, 2009], this learning problem has received surprisingly little attention so far, even though it has many interesting applications. One of the most natural applications comes in analyzing the outcomes of political elections, where the population of all voters in an electoral district is known, but only the total number of votes per party in each district is revealed. However, from an analysis of this data, e.g. the dependence of votes on variables such as income or household types, can show up interesting connections, and may be used to uncover election fraud when outliers from this model are uncovered.

Another interesting application comes from privacy-preserving data mining. In some settings, revealing the label of the observations may imply serious privacy concerns. For example, in medical research following the outbreak patterns of a new type of influenza virus is an important task, but revealing which patient actually got infected with the new virus may be viewed as information that is confidential between him and his treating physician. However, outbreak frequencies in certain risk groups are usually anonymous data, such that they are not sensitive information.

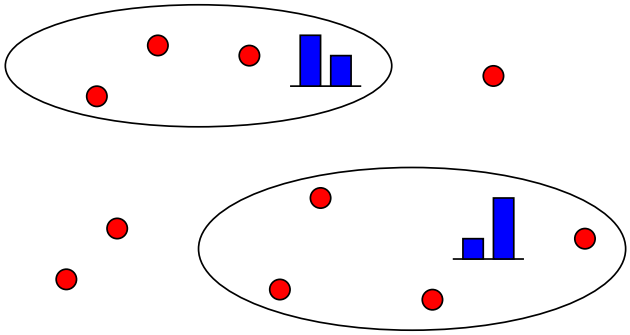


Figure 1: Classifier estimation from group probabilities

To give another example, in fraud detection it is common practice to apply machine learning to fraud / non-fraud data. While this seems to be straight-forward, in practice labeling some person as a fraudster has serious legal implications, in particular when this data is given to a third person for analysis. Even if it is clear that a person has not paid for some merchandise or service he received, there may be perfectly legal reasons not to do so. Hence, it may only be legally safe to label someone as a fraudster if he was convicted by a court of law.

In the end, storing only risk probabilities over small groups of people may be the legally advisable way in these cases. To put it more plainly, the difference between fraud labels for observations and group probabilities in this case translate to the difference between the statements *this person is a fraudster* and the much less aggressive *in this group of 5 people the risk probability is 20%*.

In this paper, we will present an algorithm for learning a classifier from group probabilities, which is based on ideas from support vector regression and classifier calibration.

The remainder of this paper is structured as follows: in the following section related work is discussed, before Section 3 introduces the new algorithm, which will be called Inverse Calibration. Section 4 empirically compares the new algorithm to existing approaches. Finally, Section 5 concludes.

[1]This paper also appears in the Proceedings of the 27th International Conference on Machine Learning (ICML 2010)

2 Related Work

In this section, we will first present related work for learning a classifier from group probabilities. We will also present existing work on the related task of estimating conditional probabilities from a given classifier, which will be relevant later on.

2.1 Estimation of a Classifier from Group Probabilities

The task of estimating a classifier from set probabilities describes the setting, where groups of unlabeled observations are given and the only information about the distribution of the labels comes from the frequencies of the labels in each group.

A method for estimating a classifier from group probabilities, the Mean Map method, has been proposed in [Quadrianto *et al.*, 2009]. The method is based on modeling the conditional class probability $p(y|x,\theta)$ using conditional exponential models:

$$p(y|x,\theta) = exp((\Phi(x,y)\theta) - g(\theta|x))$$

with a normalizing function g. The parameter θ of the model is estimated by taking the known observation means of the groups and inferring from them and the known class frequencies per group the example means given classes.

[Quadrianto *et al.*, 2009] defines the learning problem with a transductive component as well, where the distribution of the labels in the test set is known. However, in this paper we do not assume that this information is known.

Algorithmically, the Mean Map method boils down to solving a convex optimization problem. While the method is defined for joint kernels on $X \times Y$, a special case exists for the case of binary classification where $k((x,y),(x',y')) = yy'k'(x,x')$. Since in this paper we are only interested in binary classification, this variant is used in the experiments.

The paper [Quadrianto *et al.*, 2009] also gives a detailed overview of other related techniques, such as methods based on kernel density estimation, discriminative sorting, or generative models and MCMC [Kueck and de Freitas, 2005]. However, it was found that none of these methods can outperform their Mean Map method, and hence they are not investigated in detail in this paper.

2.2 Estimating Conditional Probabilities

Given a binary classification task described by an unknown probability distribution $P(X,Y)$ on an input space X and a set of labels $Y = \{-1,1\}$, a probabilistic classifier is a function $f_{prob} : X \to [0,1]$ that returns an estimate of the conditional class probability, i.e.

$$f_{prob}(x) \approx P(Y = 1|x).$$

A standard approach to probabilistic classification is to calibrate a numerical classifier. That is, for a numerical classification function

$$cl(x) = sign(f_{num}(x))$$

the task is to find an appropriate scaling function $\sigma : \mathcal{R} \to [0,1]$ such that

$$\sigma(f_{num}(x)) \approx P(Y = 1|x)$$

holds.

A comparative study by [Niculescu-Mizil and Caruana, 2005] revealed that Platt Calibration [Platt, 1999] and Isotonic Regression [Zadrozny and Elkan, 2002] are the most effective probabilistic calibration techniques for a wide range of classifiers.

Platt Calibration [Platt, 1999] was originally introduced for scaling Support Vector Machine (SVM) outputs, but has been shown to be efficient for many other numerical decision functions as well [Niculescu-Mizil and Caruana, 2005]. Based on an empirical analysis of the distribution of SVM decision function values, Platt suggests to use a scaling function of the form

$$\sigma_{Platt}(f(x)) = \frac{1}{1 + exp(-Af(x) + B)}.$$

The parameters A and B are optimized using gradient descent to minimize the cross-entropy error

Isotonic Regression [Zadrozny and Elkan, 2002] assumes a monotonic dependency between the decision function and the conditional class probabilities and finds a piecewise constant, monotonic scaling function that minimizes the quadratic loss by making use of the pair-adjacent violators algorithm [Ayer *et al.*, 1955].

Other probabilistic calibration techniques have also been proposed in the literature, for example Softmax Scaling, Binning, or calibration by Gaussian modeling of the decision function.

A finding that holds particular significance for our approach is that often even very trivial calibration techniques without an elaborate parameter estimation procedure can produce reasonably good probability estimates [Rüping, 2004].

3 The Algorithm

Problem formulation: Let $P(X,Y)$ be a fixed, but unknown probability distribution and let $(x_1,y_1),\ldots,(x_n,y_n) \subset X \times \{-1,1\}$ be drawn i.i.d. from P. Assume we are given m subsets of $(x_1,\ldots,x_n)$, where we identify the k-th subset by the set of its indices $S_k = \{i_{k,1},\ldots,i_{k,|S_k|}\}$. Let $p_k = |\{i \in S_k : y_i = 1\}|/|S_k|$ be the estimate of the conditional class probability $P(Y = 1|S_k)$. The goal is to find a classifier $f : X \to \{-1,1\}$ with minimal error according to P, given only the $x_1,\ldots,x_n$, the $S_1,\ldots,S_m$ and the $p_1,\ldots,p_m$ are known.

Inversion of Class Probability Estimation: Our approach is to invert the process of estimating conditional class probabilities from Section 2.2. In conditional class probability estimation a classifier f is trained first, and then a scaling function σ is fitted such that $\sigma(f(x))$ is a good estimate of $P(Y = 1|x)$.

We instead start with given probability estimates p, fix a scaling function σ, apply this inverse scaling function and train an SVM to predict the values $\sigma^{-1}(p)$. This approach is partly motivated by [Rüping, 2004], which shows that even very trivial probabilistic scaling functions without an elaborate parameter fitting procedure can give reasonable estimates of p.

In our algorithm, we use the scaling function

$$p = \sigma(y) = \frac{1}{1 + exp(-y)}$$

which can be seen as a special case of Platt scaling [Platt, 1999] with $A = 1$ and $B = 0$. In particular, we will make use of its inverse

$$y = \sigma^{-1}(p) = -\log(\frac{1}{p} - 1).$$

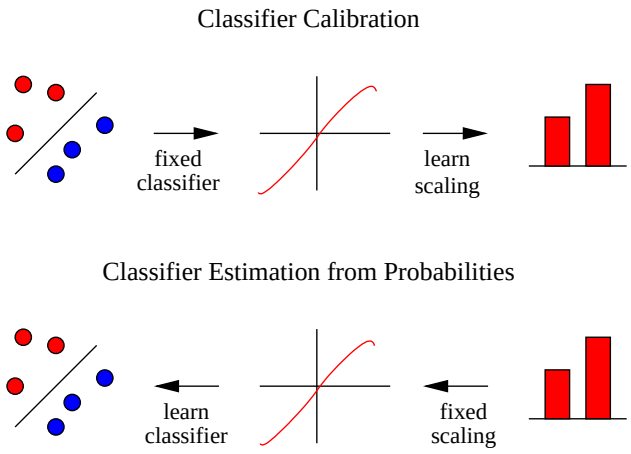


Figure 2: Classifier estimation from group probabilities by inverting the classifier calibration process.

To simplify notation, in the following, we use p and y, or p_i and y_i, interchangeably and always imply $p = \sigma(y)$. In order to avoid undefined values of y, we clip p to the interval $[\varepsilon, 1-\varepsilon]$, where ε is a parameter defining the minimum required precision of the estimate. A reasonable choice is to take $\varepsilon = 1/\#\text{examples}$.

Our goal is to estimate a linear classification function $f(x) = wx + b$. In order to classify well, we require it to predict y well, which in turn implies that $\sigma \circ f$ is a good estimate of p. However, in our problem we are not given estimates of p for every observation x, but only for sets S of observations. In particular, depending on the construction of S, the optimal class probability estimates of the single observations in S may very much vary around their average p. To circumvent this problem, we only require that f predicts y well on average:

$$\forall i : \frac{1}{|S_i|} \sum_{j \in S_i} (wx_j + b) \approx y_i.$$

We can now formally define the learning task formally in the spirit of Support Vector Regression [Vapnik, 1998], which will in particular allow a kernelization later.

Primal Problem:

$$\begin{aligned} \frac{1}{2}||w||^2 + C\sum_{i=1}^{m}(\xi_i + \xi_i^*) &\rightarrow \min \\ s.t. & \\ \forall_{i=1}^m : \xi_i, \xi_i^* &\geq 0 \\ \forall_{i=1}^m : \frac{1}{|S_i|}\sum_{j \in S_i}(wx_j + b) &\geq y_i - \varepsilon_i - \xi_i \\ \forall_{i=1}^m : \frac{1}{|S_i|}\sum_{j \in S_i}(wx_j + b) &\leq y_i + \varepsilon_i + \xi_i^* \end{aligned}$$

The formulation requires to both minimize the complexity of the model and to try to keep the class probability estimate of S_i close to p_i, where the maximum tolerable error is defined by ε_i. Note that to keep the optimization problem in the form of a quadratic problem, the error is defined with respect to y_i and not p_i, which is the true target. We will fix this inconsistency in the following.

Usually in Support Vector Regression one would use a constant value for the required precision, i.e. $\forall i : \varepsilon_i = \varepsilon$. However, in this case the goal is not to estimate y_i but p_i, such that instead of setting a precision limit on y we actually require

$$\begin{aligned} & p_i - \varepsilon \leq \sigma(y_i) \leq p_i + \varepsilon \\ \Leftrightarrow \quad & p_i - \varepsilon \leq \frac{1}{1 + exp(-y_i)} \leq p_i + \varepsilon \\ \Leftrightarrow \quad & -\log(\frac{1}{p_i - \varepsilon} - 1) \leq y_i \leq -\log(\frac{1}{p_i + \varepsilon} - 1) \end{aligned}$$

A Taylor expansion of order 1 of the function $p \mapsto -\log(\frac{1}{p+\varepsilon} - 1)$ around the point $p = p_i$ yields

$$\begin{aligned} -\log(\frac{1}{p_i + \varepsilon} - 1) &\approx -\log(\frac{1}{p_i} - 1) + \frac{\varepsilon}{p_i(1-p_i)} \\ &= y_i + \frac{\varepsilon}{p_i(1-p_i)}. \end{aligned}$$

and hence we set

$$\varepsilon_i = \frac{\varepsilon}{p_i(1-p_i)}.$$

Dual Problem: It is straightforward to prove that the primal problem can be efficiently solved in its dual form, which is

$$\begin{aligned} & \tfrac{1}{2}\sum_{i,j=1}^{m} \tfrac{(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)}{|S_i||S_j|} \sum_{i' \in S_i, j' \in S_j} K(x_{i'}, x_{j'}) \\ & + \sum_{i=1}^{m} (\alpha_i(\varepsilon_i - y_i) + \alpha_i^*(\varepsilon_i + y_i)) \rightarrow \min \end{aligned}$$

$s.t.$

$$\begin{aligned} \textstyle\sum_{i=1}^{m}(\alpha_i - \alpha_i^*) &= 0 \\ \forall_{i=1}^m : 0 \leq \alpha_i, \alpha_i^* &\leq C. \end{aligned}$$

where K is a kernel function. The minimization can be carried out by a standard solver for quadratic optimization problems.

In the following, this approach will be called Inverse Calibration.

4 Experiments

We compared the new approach, called Inverse Calibration, to learning classifiers from set probabilities empirically on 12 data sets from the UCI machine learning repository [Asuncion and Newman, 2007]. Table 1 lists the data sets that were used. To construct probability examples, we picked different set sizes k and randomly partitioned the original data sets into sets of size k (plus one set of size $< k$, if necessary). Values of $k = 2, 4, 8, 16, 32$ and 64 were chosen in the experiments. We performed tests with linear kernels, radial basis kernels with parameters $\gamma = 0.01, 0.1$ and 1 and polynomial kernels with degrees 2 and 3. Hence, in total $12 * 6 * 6 = 432$ experiments were performed. A 10-fold cross-validation was executed in each experiment. For tuning the parameters of the methods, an internal cross-validation loop was applied in each training phase.

As performance measure, we want to use the accuracy of predicting the labels in the test set. That is, we assume that while set probabilities are given in the training examples, the ultimate goal is to induce a classifier that accurately predicts the labels. In order for the analysis to be independent of the default error rate in each data sets $\mathcal{D}$, we use the accuracy of a method $\mathcal{M}$ relative to the accuracy that can

Table 1: Datasets used in the Experiments.

DATA SET	SIZE	DIMENSION
HEART-C	303	23
PRIMARY-TUMOR	339	24
IONOSPHERE	351	34
COLIC	368	61
VOTE	435	17
SOYBEAN	683	84
CREDIT-A	690	43
BREAST-W	699	10
DIABETES	768	9
VEHICLE	846	19
ANNEAL	898	64
CREDIT-G	1000	60

be achieved by a standard classification SVM that has full access to the labels on the training set as our performance measure of choice:

$$accuracy_{rel}(\mathcal{M}, \mathcal{D}) := \frac{accuracy(\mathcal{M}, \mathcal{D})}{accuracy(\text{full SVM}, \mathcal{D})}$$

We compared the Inverse Calibration algorithm with the Mean Map method which has been proven to perform superior to all competing approaches in [Quadrianto *et al.*, 2009]. In order to find out how much of the performance of the Inverse Calibration method is due to the general properties of SVMs, we compare our method against simpler approaches of applying SVMs to the problem of learning from probabilities. The following trivial variants are included in our tests:

`Reg`: directly predicting the transformed probabilities of each example using a regression SVM. The same label y was used for each element of a set.

`RegS`: directly predicting the transformed probabilities using a regression SVM, but using only the mean of the vectors in each set as an example (i.e., one example per set).

`Class`: directly predicting the label of each example, using the label 1 for every example in a set S iff the probability of S is higher than the default probability in the complete data set.

`ClassS`: same as `Class`, but taking only the mean of the vectors in each set as an example (i.e., one example per set).

Table 2 list the number of wins of each method in the columns against each method in the row, and against all other methods in total, over all trials. It can be seen that Inverse Calibration and the Mean Map method are clearly superior to the trivial methods: in a direct comparison, the trivial methods lose against the more advanced ones in at least 85% of all trials. In only 6% of all cases, a trivial method outperforms the other methods. Hence, in the following, only those two methods will be compared in more detail.

4.1 Dependency of the Performance on k

Figure 3 shows the relative accuracies of the Inverse Calibration versus the Mean Map method over all 432 tests. Both approaches generally achieve high relative accuracies, it can be seen that in most of the trials at least 70% of the accuracy of a classification SVM with full information was achieved.

To show up an interesting structure in the results, trials with low values of k, namely $k = 2, 4, 8$, were plotted as blue triangles, while trials with high values of k, namely $k = 16, 32, 64$, are plotted as red circles. For low k, i.e. for many sets with few observations per set, both approaches seem to perform roughly similar (see below for detailed statistical tests), and also often not much worse than the standard SVM. On the other hand, for high k, i.e. in a situation with less information, it can be seen that Inverse Calibration frequently outperforms the Mean Map method.

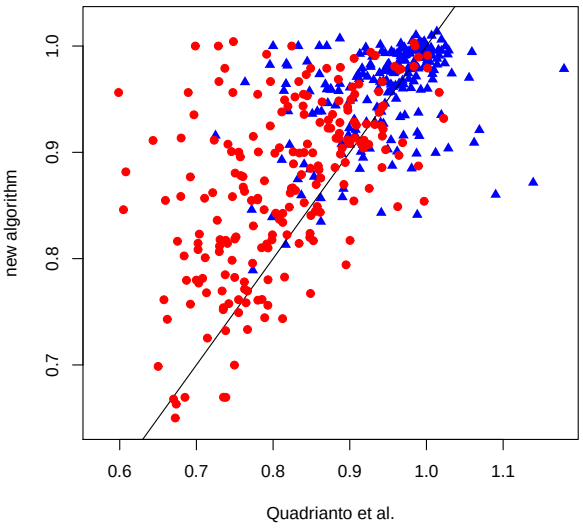


Figure 3: Relative Accuracy of Inverse Calibration (vertical axis) vs. Relative Accuracy of Mean Map (horizontal axis) over all tests. Blue triangles depict test with low values of k, while red circle depict trials with high k.

To investigate the dependency on k in detail, Figure 4 shows boxplots of the relative accuracy of Inverse Calibration minus the relative accuracy of the Mean Map method over all trials for each k. It can be seen that while generally Inverse Calibration performs better for all k (mean values are positive), the difference become especially pronounced the larger the k.

The same effect can be seen in Figure 5, which plots the actual relative accuracies of the two methods over all k. In addition, the relative accuracy of the best trivial method, `classS`, is also plotted. Again, while the relative accuracies decrease with increasing k, the gap between the different methods widens.

4.2 Dependency of the Performance on the Kernel

Finally, we are interested in the effect of the kernel function. Figure 6 shows boxplots of the relative accuracy of the Inverse Calibration method minus the relative accuracy of the Mean Map method over all trials for each kernel. It can be seen that the results are quite stable, with a slightly better performance for the linear kernel. However, the RBF kernel with parameter $\gamma = 1$ shows a high variance.

The explanation of the erratic performance of this kernel can be found in Figure 7, which shows the actual accuracies

Table 2: Comparison of methods over 432 trials. Table lists the number of wins of the method in a column against the method in a row. Last row compares the method in column against all other methods.

METHOD	INV. CALIBRATION	MEAN MAP	REG.	CLASS.	REG. SETS	CLASS. SETS
INVERSE CALIBRATION	-	139	8	13	10	33
MEAN MAP	292	-	13	52	15	62
REGRESSION	423	419	-	331	150	369
CLASSIFICATION	416	380	101	-	111	277
REGRESSION OVER SETS	422	417	36	321	-	360
CLASSIFICATION OVER SETS	395	370	58	149	66	-
ALL METHODS	268	132	0	6	1	19

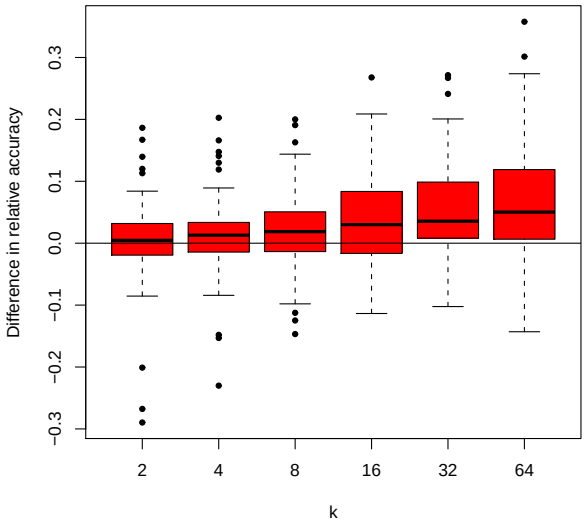


Figure 4: Boxplot of Difference in Relative Accuracy of Inverse Calibration vs. Mean Map over all k. Higher values imply better performance of Inverse Calibration.

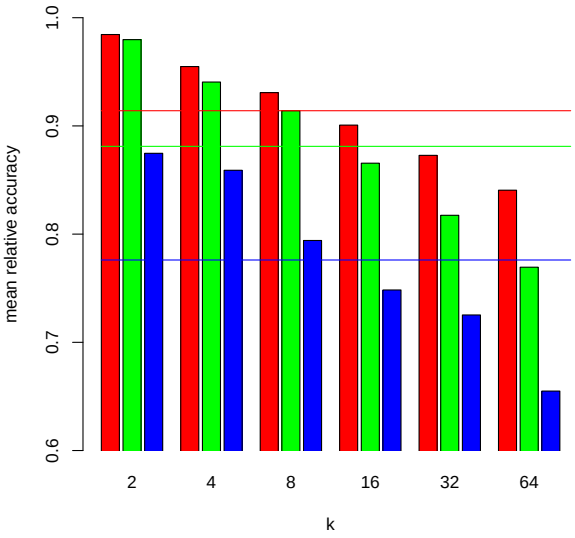


Figure 5: Relative Accuracy of Inverse Calibration (left bar, red) vs. Mean Map (middle bar, green) vs. Classification on Sets (right bar, blue) over all k. Horizontal lines show averages over all k.

of the methods. It can be seen that this kernel on the average shows a worse performance than the other ones, which is possibly due to overfitting the training data. As a consequence, random variations have a much higher influence on the performance of the learners in this case.

4.3 Overall Results

Table 3 shows a detailed comparison of Inverse Calibration with the Mean Map method over all kernels and all values of k. In total, the Inverse Calibration method outperforms Mean Map in 28 of the experiments, performs equally well on 4 and is worse on another 4 experiments (note that each experiment is a test over 12 data sets). A Wilcoxon signed rank test, as suggested by [Demsar, 2006], confirms the statistical significance of the results. Over all trials, a p-value of $6.91e-07$ is achieved, which confirms that the new Inverse Calibration method outperforms the Mean Map method. Further, the Inverse Calibration method performs particularly well for the linear kernel and RBF kernels with low γ.

Vice versa, it can be seen that the new approach is only significantly outperformed for the RBF kernel with parameter $\gamma = 1$ and $k = 2, 4$. However, as has been already discussed in Section 4.2, this kernel function exhibits a low performance on the average and hence is not of a particularly high importance compared to the other kernels.

5 Conclusions and Future Work

Estimating classifiers from group probabilities is an important learning task with many practical applications. However, it has only recently begun to receive attention in the research community.

In this paper, we have presented a new algorithm for estimating a classifier from group probabilities based on support vector regression and inverse classifier calibration. A detailed comparison of the new Inverse Calibration algorithm with the best previously known approach of [Quadrianto *et al.*, 2009] has revealed that the new algorithm performs significantly better, in particular in the case of linear kernels and high compression factors, i.e. high number k of observations per group. In all other cases, both approaches have been shown to exhibit comparable performance. Algorithmically, the Inverse Calibration method is a quadratic optimization problem, for which efficient solvers exists, while the Mean Map method has to be optimized with more general solvers.

While the new method works only for binary classification, Quadrianto's approach is also defined for an arbitrary

Table 3: Comparison of Inverse Calibration vs. Mean Map for each k and kernel. Table list the number of wins/ties/losses and the p-value of a Wilcoxon signed rank test of the hypothesis that the Inverse Calibration method is better than Mean Map. Results that are significant at the 10% level are printed in bold.

K	DOT	RBF(0.01)	RBF(0.1)	RBF(1)	POLY(2)	POLY(3)	ALL
2	**11/0/1**	7/2/3	5/0/7	3/0/9	8/0/4	6/0/6	40/2/30
	p = 0.005	*p = 0.130*	*p = 0.782*	*p = 0.989*	*p = 0.118*	*p = 0.535*	*p = 0.351*
4	**10/0/2**	9/0/3	8/0/4	4/0/8	6/0/6	7/0/5	**44/0/28**
	p = 0.015	*p = 0.156*	*p = 0.143*	*p = 0.893*	*p = 0.333*	*p = 0.465*	***p = 0.083***
8	**10/0/2**	9/0/3	9/0/3	7/0/5	9/0/3	5/0/7	**49/0/23**
	p = 0.034	*p = 0.130*	*p = 0.143*	*p = 0.602*	*p = 0.130*	*p = 0.688*	***p = 0.072***
16	**9/0/3**	**10/0/2**	**9/0/3**	**7/0/5**	6/1/5	6/0/6	**47/1/24**
	p = 0.056	***p = 0.056***	***p = 0.070***	***p = 0.050***	*p = 0.302*	*p = 0.512*	***p = 0.002***
32	**11/0/1**	**11/0/1**	**9/0/3**	**10/0/2**	**9/1/2**	8/0/4	**58/1/13**
	p = 0.018	***p = 0.027***	***p = 0.079***	***p = 0.004***	***p = 0.092***	*p = 0.235*	***p < 1e-3***
64	**10/0/2**	9/0/3	**9/0/3**	**10/0/2**	**8/0/4**	**9/0/3**	**55/0/17**
	p = 0.015	*p = 0.143*	***p = 0.070***	***p = 0.003***	***p = 0.087***	***p = 0.044***	***p < 1e-3***
	61/0/11	**55/2/15**	**49/0/23**	**41/0/31**	**46/2/24**	41/0/31	**293/4/135**
	p < 1e-3	***p = 0.027***	***p = 0.020***	***p = 0.044***	***p = 0.037***	*p = 0.275*	***p < 1e-3***

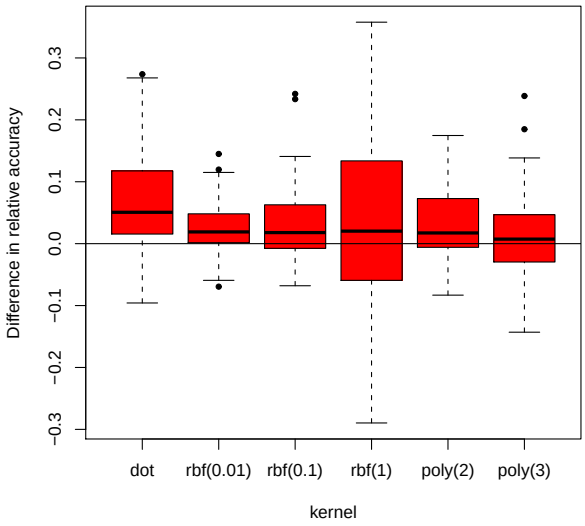


Figure 6: Difference in Relative Accuracy of Inverse Calibration vs. Mean Map over all kernels. Higher values imply better performance of Inverse Calibration.

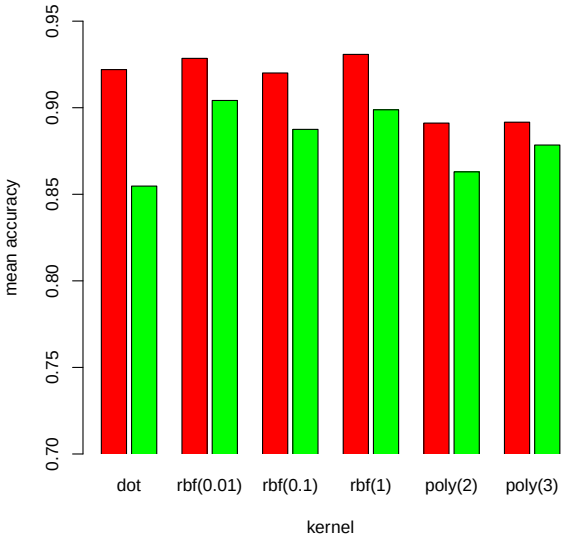


Figure 7: Comparison of the accuracy of Inverse Calibration (left bar, red) vs. Mean Map (right bar, green) over all kernels. Higher values imply better performance of Inverse Calibration.

number of classes. It would be interesting to see if it is possible to extend the new approach to multiple classes, for example by making use of ideas from algorithms for multiclass SVMs [Duan and Keerthi, 2005].

A practically very interesting direction for future work lies in taking the construction process of groups into account. In this paper, we have taken an i.i.d. assumption, which is reasonable when one does not know otherwise. However, in situations like privacy-preserving data mining, where full information is available to one party, but not the second party that is analyzing the data, both parties could still agree on a process that tries to set up the groups in a way to both guarantees data privacy and allow for an effective classifier estimation under these constraints. Such a process could, for example, be built up upon ideas from active learning [Tong and Koller, 2000].

References

[Asuncion and Newman, 2007] A. Asuncion and D.J. Newman. UCI machine learning repository. `http://www.ics.uci.edu/~mlearn/`, institution = University of California, Irvine, School of Information and Computer Sciences, 2007.

[Ayer *et al.*, 1955] M. Ayer, H. Brunk, G. Ewing, W. Reid, and E. Silverman. An empirical distribution function for sampling with incomplete information. *Annals of Mathematical Statistics*, 5:641–647, 1955.

[Demsar, 2006] Janez Demsar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006.

[Duan and Keerthi, 2005] Kai-Bo Duan and S. Sathiya Keerthi. Which is the best multiclass svm method? an

empirical study. In Nikunj C. Oza, Robi Polikar, Josef Kittler, and Fabio Roli, editors, *Proc. Multiple Classifier Systems (MCS 2005)*, volume 3541 of *LNCS*, pages 278–285. Springer, 2005.

[Kueck and de Freitas, 2005] H. Kueck and N. de Freitas. Learning about individuals from group statistics. In *Uncertainty in Artificial Intelligence (UAI)*, page 332339, Arlington, Virginia, 2005. AUAI Press.

[Niculescu-Mizil and Caruana, 2005] Alexandru Niculescu-Mizil and Rich Caruana. Predicting good probabilities with supervised learning. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 625–632, 2005.

[Platt, 1999] John Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In A. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*. MIT Press, 1999.

[Quadrianto *et al.*, 2008] Novi Quadrianto, Alex J. Smola, Tibrio S. Caetano, and Quoc V. Le. Estimating labels from label proportions. In W. Cohen, A. McCallum, , and S. Roweis, editors, *Proceedings of the 25th Annual International Conference on Machine Learning (ICML 2008)*, page 776783. Omnipress, 2008.

[Quadrianto *et al.*, 2009] Novi Quadrianto, Alex J. Smola, Tibrio S. Caetano, and Quoc V. Le. Estimating labels from label proportions. *Journal of Machine Learning Research*, 10:2349–2374, Oct 2009.

[Rüping, 2004] Stefan Rüping. A simple method for estimating conditional probabilities in SVMs. In A. Abecker, S. Bickel, U. Brefeld, I. Drost, N. Henze, O. Herden, M. Minor, T. Scheffer, L. Stojanovic, and S. Weibelza hl, editors, *LWA 2004 - Lernen - Wissensentdeckung - Adaptivität*. Humboldt-Universität Berlin, 2004.

[Tong and Koller, 2000] S. Tong and D. Koller. Restricted bayes optimal classifiers. In *Proceedings of the 17th National Conference on Artificial Intelligence (AAAI)*, 2000.

[Vapnik, 1998] V. Vapnik. *Statistical Learning Theory*. Wiley, Chichester, GB, 1998.

[Zadrozny and Elkan, 2002] Bianca Zadrozny and Charles Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 694–699, 2002.

Fast-Ensembles of Minimum Redundancy Feature Selection

Benjamin Schowe and **Katharina Morik**
Technische Universität Dortmund
{schowe,morik@ls8.cs.tu-dortmund.de}

Abstract

Finding relevant subspaces in very high-dimensional data is a challenging task not only for microarray data. The selection of features must be stable, but on the other hand learning performance is to be increased. Ensemble methods have succeeded in the increase of stability and classification accuracy, but their runtime prevents them from scaling up to real-world applications. We propose two methods which enhance correlation-based feature selection such that the stability of feature selection comes with little or even no extra runtime. We show the efficiency of the algorithms analytically and empirically on a wide range of datasets.

Potenzial des Data Mining für Ressourcenoptimierung mobiler Geräte im Krankenhaus

Rene Schult, Bastian Kurbjuhn
Otto-von-Guericke-Universität
39106 Magdeburg, Deutschland
schult@ovgu.de
bastian.kurbjuhn@st.ovgu.de

Abstract

Computerunterstütztes Ressourcenmanagement in Krankenhäusern wird, nicht zuletzt durch den Kostendruck, immer wichtiger. Wir besprechen, wie mit Data Mining Technologien das Ressourcenmanagement für mobile medizinische Geräte in einem Krankenhaus unterstützt werden kann und welches Potenzial in der Auswertung von Nutzungsdaten für ein Krankenhaus bestehen.

1 Einleitung

In Krankenhäusern gibt es viele mobile medizinische Geräte, die für die Behandlung am Patienten benötigt werden. In einem Krankenhaus mittlerer Größenordnung summiert sich der Anschaffungspreis dieser Geräte auf mehrere Hunderttausende von Euro oder gar darüber. Dies stellt einen erheblichen Anteil am gebundenen Kapital eines Krankenhauses dar. Durch den Kostendruck der Krankenhäuser und der Anforderung, mit den eingesetzten Ressourcen effizienter umzugehen, ergeben sich aus der Sicht des Ressourcenmanagements sowie aus der Literatur heraus u.a. folgende Problemstellungen, die auch beim Facility Management in Krankenhäusern immer mehr an Bedeutung gewinnen: (i) Einhaltung von gesetzlichen Vorgaben, etwa zur Dokumentation der Nutzung oder zur Wartung der Geräte, (ii) Einhaltung von Vorgaben zur Verfügbarkeit der Geräte oder zur Qualität der Leistung, ([Pocsay and Distler, 2009]; [Mauro *et al.*, 2010]) (iii) Minimierung der Investitionskosten und der laufenden Kosten ([Salfeld *et al.*, 2009]; [Mauro *et al.*, 2010]), (iv) Minimierung des Wartungsaufwands und somit auch der Wartungskosten ([Salfeld *et al.*, 2009]; [Mauro *et al.*, 2010]).

Im Folgenden schildern wir anhand eines Fallbeispiels die Aspekte, die beim Ressourcenmanagement von mobilen Geräten im Krankenhaus zu berücksichtigen sind, und erläutern dabei das Potenzial der Erfassung und Analyse von Daten zur Gerätenutzung. Das hier geschilderte Problem bezieht sich derzeit auf die tragbaren Infusionsspritzen und Infusionspumpen, kann aber inhaltlich sicher auf weitere, wenn nicht gar alle mobilen medizinischen Geräte übernommen werden.

Es besteht die Problematik, dass das technische Personal jedes Gerät für die Erfüllung von Wartungsaufgaben oder anderen Servicedienstleistungen finden muss, bzw. wissen muss, ob es derzeit gerade bei einem Patienten im Einsatz ist oder nicht. Die derzeitig gesetzlich vorgeschriebenen Wartungsintervalle für die oben genannten Geräte beträgt ein Jahr.

Das Problem des Auffindens eines Gerätes hat auch das Schwesternpersonal, wenn es ein freies Gerät für einen neuen Patienten sucht. Für die Geräte existiert je Station meist ein zentraler Abstellplatz, wobei der je nach Station sehr unterschiedlich sein kann. Um nun ein freies Gerät für einen neuen Patienten oder für die Wartung zu finden, wird derzeit der Lagerraum der Station aufgesucht. Ist das Gerät dort nicht vorhanden, wird die Station nach dem Gerät abgesucht. Dieses Vorgehen ist gerade für das technische Personal sehr zeitintensiv, denn einerseits muss das technische Personal den jeweiligen Abstellort der Station kennen und falls das gesuchte Gerät dort nicht vorhanden ist, die gesamte Station danach absuchen und hoffen, dass es gerade nicht bei einem Patienten im Einsatz ist. Durch dieses bisherige Vorgehen und der Notwendigkeit aus medizinischer Sicht, ausreichend mobile medizinische Geräte bereitzuhalten, ist die derzeitige Situation so, dass viel mehr Geräte der unterschiedlichen Typen, in diesem Beispiel Infusionspumpen und -spritzen, bereit gehalten werden, als tatsächlich benötigt werden. Wenn man bedenkt, dass eines dieser Geräte zwischen 1000 und 1800 Euro kostet und man die aktuellen Zahlen des Krankenhauses sich vor Augen führt, dass 151 Infusionspumpen und 286 Infusionsspritzen derzeit in Benutzung sind, erkennt man, dass allein nur bei diesen beiden Gerätetypen mehr als 550.000 Euro als gebundenes Kapital im Bestand sind.

Unser Lösungsansatz umfasst folgende Komponenten: (1) Ortung der mobilen medizinischen Geräte, (2) online Erfassung der Bewegungsdaten zu den Geräten durch automatische Auswertung der Ortungsdaten, (3) Ableitung von nutzungsrelevanten Informationen (Ereignisse) aus den Bewegungsdaten und (4) Auswertung der Nutzungsdaten zur Berechnung der tatsächlichen Verfügbarkeit und des tatsächlichen Bedarfs für ein jedes Gerät. Darauf aufbauend soll die Optimierung inkl. Vorhersagen stattfinden. Durch die geeignete Nutzung technischer Möglichkeiten soll die Ortung der mobilen medizinischen Geräte vereinfacht und deren Nutzungsstatus der Geräte abgebildet werden. Basierend darauf soll danach die Nutzung der mobilen medizinischen Geräte (Infusionspumpen und -spritzen) mit Hilfe von Data Mining Methoden analysiert werden, um eine optimalere Nutzung der Geräte zu ermöglichen und im letzten Schritt die Möglichkeiten der Reduzierung des Gerätebestandes zu untersuchen. Der Einsatz von Data Mining Methoden bietet sich vor allem unter der Nebenbedingung an, dass das medizinische Personal für die Ortung und Nutzungsbestimmungen der medizinischen Geräte ihre bestehenden Arbeitsprozesse möglichst wenig bzw. gar nicht anpassen will und kann.

Durch diese Optimierungsschritte soll eine detailliertere

Planung des Bestandes der medizinischen Geräte und deren Auslastung möglich werden, was u.a. zur Verringerung des gebundenen Kapitals führen kann.

2 Relevante Literatur

Das Ressourcenmanagement (auch Facility Management genannt) wird schon seit langem in der Literatur diskutiert. Zunehmend findet das computerunterstützte Ressourcenmanagement (CAFM) Betrachtung ([Nävy, 2006]). Die German Facility Management Association (GEFMA) hat in ihrem Arbeitskreis für Facility Management im Krankenhaus unter anderem herausgestellt, dass durch die Einführung der DRGs nach professionelleren Vorgehensweisen und Optimierungspotenzialen ([Odin, 2010]) in Krankenhäusern gesucht wird. In ([Köchlin, 2004]) verdeutlicht Köchlin den Nutzen von Facility Management im Krankenhaus. In ([Salfeld *et al.*, 2009]) stellen Salfeld et al. heraus, dass vor allem die nicht klinischen Funktionen Kosteneinsparungspotenziale bieten, zu denen auch das Facility Management gehört. In ([Pocsay and Distler, 2009]) und in ([Mauro *et al.*, 2010]) betrachten die Autoren vor allem das Potenzial der IT, um mögliche Einsparungspotenziale zu nutzen, bzw. die Prozesse in einem Krankenhaus zu verbessern. Der Schwerpunkt liegt bei beiden jedoch eher auf dem Austausch der Daten, was u.a. durch SOA ermöglicht werden soll, als auf der Optimierung von Prozessen und Ressourcen.

Data Mining Methoden kamen bei Alapont et. al. in [Alapont *et al.*, 2004] bei der Auslastungsprognose von Betten in spanischen Krankenhäusern zum Einsatz. Dort wurden die statistischen Methoden LinearRegression, LeastMedSq, SMOreg, MultilayerPerception, Kstart, LWL, Tree DecissionStump, Tree M5P und IBK miteinander verglichen. Die Methoden LinearRegression und Tree M5P eigneten sich dort am besten, um die Prognosefehlerrate zu reduzieren.

Die Betrachtung von tragbaren medizinischen Geräten als Ressource des Krankenhauses in Kombination mit dem computerunterstützten Ressourcenmanagement wurde bisher in der Literatur nicht betrachtet.

3 Grundlagen

Für ein effektives Ressourcenmanagement werden Daten über die Nutzung der zu betrachtenden Geräte benötigt. Gerade bei mobilen Geräten ist die Beschaffung der Daten über die Nutzung und deren Lage nicht so einfach, durch deren Eigenschaft, dass die Geräte mobil nutzbar sind. In unserem Projekt erfolgt die Ortung mittels WLAN Technologie. Ein WLAN Chip wird am zu überwachenden Gerät angebracht und meldet sich in regelmäßigen Abständen bei einem Server über verschiedene AccessPoints. Durch spezielle Software lässt sich die Lage eines Gerätes ermitteln. Diese Lagepositionen in Kombination mit einem Zeitstempel werden in einer Datenbank gespeichert. Die Datenbankstruktur ist in Tabelle 1 dargestellt.

In der Tabelle 1 werden zu jedem WLAN Chip (TagID) die dazugehörigen Parameter wie X und Y Koordinaten der aktuellen Position sowie Mapping Parameter dafür eingetragen. Des Weiteren werden der aktuelle Zeitstempel des Eintrags und die Sendequalität des WLAN Chips gespeichert. Somit stellt die Datenbank eine Art Log-File für die einzelnen WLAN-Chips und deren aktuellen Positionen da. Mittels Data Mining Techniken zur Log-File Analyse lassen sich die Daten für einzelne Geräte gruppieren und heraus filtern. Durch Auswertung dieser örtlichen Daten zu einem speziellen Gerät und dem Nutzen von räumlichen Informationen zum Gebäude, lassen sich Daten zur Nutzung eines Gerätes ermitteln. Dadurch stehen für jedes medizinische Gerät Daten über deren Position und deren Nutzung in einem zeitlichen Verlauf zur Verfügung. Durch diese zeitlichen Daten lassen sich für jedes einzelne Gerät, aber auch für jede Station oder auch das gesamte Krankenhaus verschiedene Kennzahlen, wie Auslastung des Gerätes, Ruhezeiten, Reparaturzeiten, zurückgelegte Wege usw. ermitteln.

4 Wissenschaftliche Fragestellungen

Die aus Business Intelligence Sicht interessanten Fragestellungen ergeben sich aus der Analyse dieser einzelnen Zeitreihen.

1. Es ist für das Krankenhaus interessant, zu analysieren, wie groß die Auslastung der einzelnen medizinischen Geräte ist. Darauf aufbauend kann man versuchen zu analysieren, worin die möglichen Unterschiede der Nutzungsgrade der einzelnen Stationen liegen. Dies bedeutet eine Klassifikation so durchzuführen, dass man unterscheiden kann, welche Eigenschaften der erfassten Daten auf eine hohe bzw. geringe Nutzung der Geräte zurückzuführen ist.
2. Aus den Nutzungsstatistiken und den erkannten Abhängigkeiten für eine hohe bzw. geringe Nutzung der Geräte kann man versuchen eine Vorhersage abzuleiten für den zukünftigen Bedarf an den medizinischen Geräten. Gerade diese Vorhersagen können ein Einsparpotenzial aus ökonomischer Sicht bedeuten, denn wenn diese Vorhersagen geringer als der derzeitige Gerätebestand ausfallen, reichen weniger Geräte für die gleiche medizinische Leistung ohne Qualitätseinbußen. Sicher sollte man dabei aber noch eine Art Puffer an Geräten einführen, um z.B. Schwankungen nach oben im Bedarf abfangen zu können.

Um die erste beschriebene Fragestellung zu lösen, ermittelt man auf den vorhandenen Daten der Datenbank und den darauf folgenden Aufbereitungen der Log-Daten der einzelnen Geräte ein deskriptives Modell auf den Daten um zwischen den Eigenschaften für die Nutzung und der Nichtnutzung unterscheiden zu können. Hierbei ist es interessant herauszufinden, ob es

1. überhaupt typische Eigenschaften für die Nutzung bzw. Nichtnutzung der Geräte gibt und
2. mit welchen Data Mining Methoden lassen diese sich am besten ermitteln bzw. die Daten in die definierten Nutzergruppen am besten unterteilen. Eignet sich dazu z.B. besser eine Regressionsanalyse oder ein Clusteringalgorithmus oder auf Klassifikationsmodelle?

Für die zweite Fragestellung ist es von starkem Interesse, ob sich qualitativ hochwertige Vorhersagen über die Nutzung der einzelnen Geräte machen lassen. Dabei kann man versuchen zwischen den einzelnen Geräten, den Geräten die einer Station zugewiesen sind oder auch den Gerätegruppen des gesamten Krankenhauses zu unterscheiden. Die Vorhersagen in qualitativ hochwertiger Weise

Datenfeld	Datentyp	Zusätze	Beispiel
Id	BIGINT UNSIGNED	NOT_NULL AUTO_INCREMENT PRIMARY_KEY	1
TagId	VARCHAR(20)	NOT_NULL	3003121670
MacAdress	VARCHAR(20)		00:00:B3:00:00:06
PosX	INT(10)	NOT_NULL	240
PosY	INT(10)	NOT_NULL	588
PosModelId	INT(10)		1
PosMapId	INT(10)		3
PosMapName	VARCHAR(100)		floor2
PosZoneId	INT(10)		17
PosZoneName	VARCHAR(100)		hall-a
PosQuality	INT(10)		99
PosTimestamp	TIMESTAMP	NOT_NULL	29.05.06 15:21

Table 1: Datenbankstruktur für die Ortung

wären eine ideale Grundlage, um das Ressourcenmanagement des Krankenhauses für diese medizinischen Geräte zu unterstützen und gleichzeitig Kosten durch die Verringerung des gebundenen Kapitals in diese Geräte zu sparen.

5 Ansatz für die Umsetzung

Aus den gewonnenen Informationen zur Ortung des Gerätes kann als Eigenschaft (Zielgröße) definiert werden, ob ein Gerät zur Zeit im Einsatz ist: Hierzu müssen jedoch sogenannte Lagerorte bestimmt werden. Nach der Datenbankdefinition setzt sich der Standort aus der PosMapId(Name) und der PosZoneId(Name) zusammen bzw. werden so die Räume abgebildet (siehe Tabelle 1). Weist man nun bestimmte Räume als (Stations-)Lagerorte aus, so kann anhand der Position des Gerätes abgeleitet werden, ob es sich zu der Zeit im Einsatz befindet. Da die Wartungen außer Haus stattfinden, wären die Geräte zu Reparaturzeiten nicht im WLAN des Krankenhauses angemeldet. Bei einem kontinuierlichen Logging bedeutet ein Fehlen eines Gerätes in der Ortungsdatenbank über einen kurzen Zeitraum, dass das Gerät sich in der Wartung befindet oder entwendet wurde. Es kann aber auch bedeuten, dass technische Störungen vorliegen. Diese Ereignisse können über die Dauer der fehlenden Protokollierung voneinander abgegrenzt werden. Da die durchschnittliche Dauer einer Geräteüberprüfung aus den Wartungsinformationen der Vergangenheit bekannt ist (und die Prüfung einem Standard folgt), kann ein erlaubter Zeitraum definiert werden, in dem das Gerät fernbleiben darf. Wird diese Zeitgrenze mitsamt einer dazugehörigen durchschnittlichen Reparaturzeit überschritten, steigt die Wahrscheinlichkeit, dass ein Fehler in der Kommunikation des Chips mit dem Access Point vorliegt. Bei einer Auswertung zur Bestimmung des Nutzungsgrades müssen die Daten, die von solchen technischen Problemen betroffen sind, bereinigt werden (also zu große Fehlzeiten sind in Frage zu stellen). In Verbindung mit dem Zeitstempel und den Verlaufsdaten bietet es sich dann an, die Auslastung einzelner Geräte zu bestimmen. Für die Ermittlung des Auslastungsgrades pro Gerät gilt:

$$Auslastungsgrad = \frac{Einsatzzeiten + Reparaturzeiten}{Betrachtungszeitraum}$$

Diese Schritte zur Ausrechnung des Auslastungsgrades und der Bereinigung der Daten bzgl. der Wartungszeiten oder technischen Probleme sind typische Schritte der Datenaufbereitung in diesem Optimierungsprojekt.

5.1 Deskriptives Modellkonzept - Clustering von Geräten mit ähnlichen Nutzungsverhalten

Nachdem nun die Auslastungsgrade der einzelnen Geräte, deren Lager- und Reparaturzeiten anhand der vorhandenen Log-Daten in der Ortungsdatenbank bestimmt werden können, muss betrachtet werden, welche Eigenschaften eine Nutzung bzw. eine Nichtnutzung der Geräte ausmachen. Folgende Einflussfaktoren können sich auf die Auslastung niederschlagen:

- **der Verbraucher des Gerätes (die Station)** Bei welcher Station ist das Gerät zuletzt im Einsatz gewesen? Die Länge des Aufenthaltes in der Station gibt Aufschluss darüber, welche Stationen gegenüber anderen Abteilungen die größeren Verbraucher sind bzw. mehr auf die Geräte als andere Stationen angewiesen sind. Wie oft greift eine Station auf ein Gerät zu? So kann ermittelt werden, ob eine Station auf bestimmte Geräte angewiesen ist und die Station Hauptnutzer dieses Gerätes ist.
- **der aktuelle Lagerort/Standort (Raumbezug)** Welcher Lagerort ist dem Gerät zugewiesen? Wenn ein Gerät einen Lagerraum hat, an dem sich in der näheren Umgebung keine Großverbraucher (Stationen mit besonders hohem Bedarf an mobilen Geräten) befinden, ist die Wahrscheinlichkeit geringer gegenüber anderen gleichen Geräten, dass sie ausgeliehen (genutzt) werden. So wird die Annahme getroffen, dass sie einen unterdurchschnittlichen Nutzungsgrad aufweisen werden als ihre Kollegen. Denn die Beschaffung eines Gerätes in einer Umgebung verbrauchsarmer Stationen ist für einen Verbraucher mit zusätzlichem (Weg-)Aufwand verbunden. Es wird unterstellt, dass solch ein Gerät den letzten Ausweg darstellt, falls keine gleichen in der näheren Umgebung für die Nachfragestation nicht in Benutzung sind.
- **die Natur des Gerätes (Gerätetyp)** Was für ein Gerätetyp liegt vor? Unter Berücksichtigung externer statistischer Erhebungen in der Gesundheitsbranche wird deutlich, dass ein Einsatz eines Gerätes auch von seiner Ausprägung abhängt. Annahme: Allge-

meine Geräte (Infusionsspritzen und -pumpen) werden häufiger verwendet als Spezialgeräte (Herzmassagegeräte/Beatmungsgeräte). Per WLAN-Chip-ID wird ein Gerät eindeutig identifiziert. Wenn bekannt ist, um was für ein Gerät es sich handelt (Speicherung der Geräteinformationen in einer weiteren Stammdatenbank oder Erweiterung der Ortungsdatenbank um das Feld Type (Gerätetyp), kann dieser Einflussfaktor selbst bestimmt werden.

Hieraus wird ersichtlich, dass sich Verbraucher, Lagerort und Gerät gegenseitig bedingen und den Auslastungsgrad beeinflussen. Es muss nun näher untersucht werden, welche Konstellationen für eine Nutzung sprechen und welche eine Lagerung verantworten.

5.2 Anwendung von Data-Mining-Verfahren

Bei der statistischen Methode werden die Kennzahlen wie Auslastungsgrad (in Bezug auf das Gerät) und Verwendungsgrad (in Bezug auf die Station) aus den Werten der Datenbank über gezielte Datenbankabfragen berechnet. Um jedoch auch Prognosen bzw. Trends, die eine Einsatzplanung der Geräte erlauben, bewerkstelligen zu können, bedarf es geeignete Methoden aus dem Bereich des Data Mining zu finden.

Auf der einen Seite bieten sich Clusteralgorithmen, wie der K-Means Algorithmus an, um die Geräte mit ähnlichen Nutzungsverhalten bzw. ähnlichen Eigenschaften zu gruppieren. Die Idee dahinter ist folgende, dass ähnliche Eigenschaften der Geräte in der Nutzungsdatenbank auch zu ähnliche Nutzungsstatistiken führen. Aus den speziellen Eigenschaften der entstandenen Cluster lassen sich dann Rückschlüsse ziehen auf unterschiedliche Bedingungen für die unterschiedlich starken Nutzungseigenschaften der Geräte. Durch die Möglichkeit vieler Noise-Punkte, die z.B. beim Bewegen der Geräte entstehen können, sind vielleicht auch Hierarchische Algorithmen oder dichtebasierte Algorithmen, wie der DB-Scan oder die incrementelle Version davon, eher zu nutzen in diesem Anwendungsfall.

Aus den Clusteringergebnissen kann man den aktuellen Stand der Nutzung der einzelnen Geräte ablesen und Schlussfolgerungen darüber ziehen, welche Eigenschaften das Nutzungsverhalten beeinflussen. Um Vorhersagen über die zukünftige Nutzung der Geräte zu machen, sind die deskriptiven Modelle nicht geeignet. Hier müssen Vorhersagemodelle, wie Entscheidungsbäume zum Einsatz kommen. Bei den Klassifikationsmodellen ist die Zielvariable die vorher ermittelte Nutzung der einzelnen Geräte. Da diese jedoch in einem Wertebereich von 0 bis zum gesamten Betrachtungszeitraum liegen kann, ist es einfacher, den Wertebereich in zwei (hoch, gering) oder gar drei (hoch, mittel, gering) Intervalle zu unterteilen, die den Grad der Nutzung angeben. Hierfür lassen sich dann verschiedene Entscheidungsbäume testen oder z.B. auch der NaiveBayes Algorithmus, um zu sehen welche Eigenschaften der Daten den einen oder anderen Nutzungsgrad beeinflussen. Danach kann man mit unterschiedlichen Testdaten, in denen man die Werte der einzelnen Eigenschaften variiert, die einzelnen Nutzungsgrade vorhersagen. Aus diesen Vorhersagen lassen sich für ein Krankenhaus Rückschlüsse ziehen, welche Eigenschaften intensiviert oder verringert werden sollen, damit die gewünschten Nutzungsgrade der mobilen Geräte erreicht werden.

Eine weitere Möglichkeit der Analyse der Daten ermöglicht die Trajectorenanalyse, wenn man die Bewegungsdaten der mobilen Geräte nicht als Noise-Punkte interpretiert sondern einfach als Daten eines Trajectories. Diese lassen sich dann auf Ähnlichkeiten gruppieren bzw. lassen sich Besonderheiten feststellen. Dann lassen sich die einzelnen Wege der Geräte analysieren und als ein mögliches Ergebnis daraus die Standorte für die Aufbewahrung optimieren. Diese Optimierung zielt darauf ab, die Laufwege des Schwesternpersonals zu verringern. Dies kann dadurch erreicht werden, in dem der Lagerort für die Geräte an einen aus den Trajectoren herausgefundenen zentraleren Ort verlagert wird.

6 Zusammenfassung

Wir haben gezeigt, dass die Ressourcenoptimierung in einem Krankenhaus viel Potenzial für den Einsatz von Data Mining Techniken bietet. Anhand der beschriebenen Problemstellung, dem Wiederfinden der mobilen Geräte, und den dargestellten Lösungsansätzen können sich die Nutzung der Ressourcen und der Aufwand für das nutzende Personal verringern lassen. Bei der Analyse der Auslastung der mobilen Geräte lassen sich beschreibende und vorhersagende Algorithmen für unterschiedliche Zielstellungen einsetzen. Die Umsetzung dieser Idee und die tatsächlichen Auswirkungen der Nutzung von Data Mining bei der Ressourcenoptimierung werden wir im Projekt untersuchen. Als Anmerkung sei kurz erwähnt, dass die hier beschriebenen Schritte derzeit in der Planung bzw. am Anfang der Umsetzung erst sind und somit noch keine experimentellen Ergebnisse vorliegen. Das Paper ist als Diskussionspapier gedacht.

References

[Alapont *et al.*, 2004] J. Alapont, A. Bella-Sanjun, C. Ferri, J. Hernndez-Orallo, J. D. Llopis-Llopis, and M. J. Ramrez-Quintana. Specialised Tools for Automating Data Mining for Hospital Management. Technical report, Universitat Politcnica de Valncia, Cam de vera s/n, 46022 Valencia, Spain, 2004.

[Köchlin, 2004] K. Köchlin. Ist der EInsatz von Facility Management im Krankenhaus notwendig? Technical report, Technische FH Berlin, 2004.

[Mauro *et al.*, 2010] C. Mauro, J.M. Leimeister, and H. Krcmar. Serviceorientierte Integration medizinischer Geräte ganzheitliche IT-Unterstützung klinischer Prozesse. *Informatik Spektrum*, April 2010.

[Nävy, 2006] Nävy. *Facility Management - Grundlagen, COmputerunterstützung, Systemeinführung, Anwendungsbeispiele*. Springer Verlag, 2006.

[Odin, 2010] S. Odin. GEFMA Arbeitskreis Facility Management im Krankenhaus - Benchmarking. Technical report, GEFMA e.V., 2010.

[Pocsay and Distler, 2009] A. Pocsay and O Distler. *Zukunftsorientierter Wandel im Krankenhausmanagement Outsourcing, IT-Nutzenpotenziale, Kooperationsformen, Changemanagement*, chapter Geschäftsprozessmanagement im Gesundheitswesen Organisation und IT wachsen zusammen. Springer Verlag, April 2009.

[Salfeld *et al.*, 2009] Salfeld, Hehner, and Wichels. *Modernes Krankenhaus Management, Konzepte und Lösungen*. Springer Verlag, 2009.

Probability Estimation and Aggregation for Rule Learning

Jan-Nikolas Sulzmann, Johannes Fürnkranz

Abstract

Rule learning is known for its descriptive and therefore comprehensible classification models which also yield good class predictions. For different classification models, such as decision trees, a variety of techniques for obtaining good probability estimates have been proposed and evaluated. However, so far, there has been no systematic empirical study of how these techniques can be adapted to probabilistic rules and how these methods affect the probability-based rankings. In this paper we apply several basic methods for the estimation of class membership probabilities to classification rules. We also study the effect of a shrinkage technique for merging the probability estimates of rules with those of their generalizations. Finally, we compare different ways of combining probability estimates from an ensemble of rules. Our results show that for probability estimation it is beneficial to exploit the fact that rules overlap (i.e., rule averaging is preferred over rule sorting), and that individual probabilities should be combined at the level of rules and not at the level of theories.

1 Introduction

The main focus of symbolic learning algorithms such as decision tree and rule learners is to produce a comprehensible explanation for a class variable. Thus, they learn concepts in the form of crisp IF-THEN rules. On the other hand, many practical applications require a finer distinction between examples than is provided by their predicted class labels. For example, one may want to be able to provide a confidence score that estimates the certainty of a prediction, to rank the predictions according to their probability of belonging to a given class, to make a cost-sensitive prediction, or to combine multiple predictions.

All these problems can be solved straight-forwardly if we can predict a probability distribution over all classes instead of a single class value. A straight-forward approach to estimate probability distributions for classification rules is to compute the fractions of the covered examples for each class. However, this naïve approach has obvious disadvantages, such as that rules that cover only a few examples may lead to extreme probability estimates. Thus, the probability estimates need to be smoothed.

There has been quite some previous work on probability estimation from decision trees (so-called *probability-estimation trees (PETS)*). A very simple, but quite powerful technique for improving class probability estimates is the use of m-estimates, or their special case, the Laplace-estimates [Cestnik, 1990]. It has been shown that unpruned decision trees with Laplace-corrected probability estimates at the leaves produce quite reliable decision tree estimates [Provost and Domingos, 2003]. A recursive computation of the m-estimate, which uses the probability disctribution at level l as the prior probabilities for level $l + 1$, was proposed in [Ferri *et al.*, 2003]. In [Wang and Zhang, 2006], a general shrinkage approach was used, which interpolates the estimated class distribution at the leaf nodes with the estimates in interior nodes on the path from the root to the leaf.

An interesting observation is that, contrary to classification, class probability estimation for decision trees typically works better on unpruned trees than on pruned trees. The explanation for this is simply that, as all examples in a leaf receive the same probability estimate, pruned trees provide a much coarser ranking than unpruned trees. In [Hüllermeier and Vanderlooy, 2009], a simple but elegant analysis of this phenomenon was provided, which shows that replacing a leaf with a subtree can only lead to an increase in the area under the ROC curve (AUC), a commonly used measure for the ranking capabilities of an algorithm. Of course, this only holds for the AUC estimate on the training data, but it still may provide a strong indication why unpruned PETs typically also outperform pruned PETs on the test set.

Despite the amount of work on probability estimation for decision trees, there has been hardly any systematic work on probability estimation for rule learning. Despite their obvious similarility, we nevertheless argue that a separate study of probability estimates for rule learning is necessary.

A key difference is that in the case of decision tree learning, probability estimates will not change the prediction for an example, because the predicted class only depends on the probabilities of a single leaf of the tree, and such local probability estimates are typically monotone in the sense that they all maintain the majority class as the class with the maximum probability. In the case of rule learning, on the other hand, each example may be classified by multiple rules, which may possibly predict different classes. As many tie breaking strategies depend on the class probabilities, a local change in the class probability of a single rule may change the global prediction of the rule-based classifier, even if the order of all local estimates is maintained.

Because of such non-local effects, it is not evident that the same methods that work well for decision tree learning will also work well for rule learning. Indeed, as we will see in this paper, our conclusions differ from those that

have been drawn from similar experiments in decision tree learning. For example, the above-mentioned argument that unpruned trees will lead to a better (training-set) AUC than pruned trees, does not straight-forwardly carry over to rule learning, because the replacement of a leaf with a subtree is a local operation that only affects the examples that are covered by this leaf. In rule learning, on the other hand, each example may be covered by multiple rules, so that the effect of replacing one rule with multiple, more specific rules is less predictable. Moreover, each example will be covered by some leaf in a decision tree, whereas each rule learner needs to induce a separate default rule that covers examples that are covered by no other rule.

The rest of the paper is organized as follows: In Section 2 we briefly describe the basics of probabilistic rule learning and discuss the used estimation techniques used for rule probabilities. In Section 3 we describe the rule learning algorithm that we used in our experiments, contrast two approaches for the generation of a probabilistic rule set, and describe how they are used for classification. Experimental results, which compare the different probability estimation techniques in these two scenarios, are described in Section 4. We then discuss four techniques for obtaining rule probabilities from a bagging ensemble (Section 5), and compare them experimentally in Section 6. In the end, we summarize our conclusions in Section 7.[1]

2 Rule Learning and Probability Estimation

This section is divided into two parts. The first one describes briefly the properties of conjunctive classification rules and of its extension to a probabilistic rule. In the second part we introduce the probability estimation techniques used in this paper. These techniques can be divided into basic methods, which can be used stand-alone for probability estimation, and the meta technique shrinkage, which can be combined with any of the techniques for probability estimation.

2.1 Probabilistic Rule Learning

In classification rule mining one searches for a set of rules that describes the data as accurately as possible. As there are many different generation approaches and types of generated classification rules, we do not go into detail and restrict ourselves to conjunctive rules. The *premise* of these rules consists of a conjunction of number of conditions, and in our case, the *conclusion* of the rule is a single class value. So a conjunctive classification rule r has basically the following form:

$$condition_1 \wedge \cdots \wedge condition_{|r|} \implies class \qquad (1)$$

The size of a rule $|r|$ is the number of its conditions. Each of these conditions consists of an attribute, an attribute value belonging to its domain and a comparison determined by the attribute type. For our purpose, we consider only nominal and numerical attributes. For nominal attributes, this comparison is a test of equality, whereas in the case of numerical attributes, the test is either less (or equal) or greater (or equal). If all conditions are met by an instance, the instance is covered by the rule ($r \supseteq x$) and the class value of the rule is predicted for the instance. Consequently, the rule is called a *covering rule* for this instance.

[1]Parts of this paper have previously appeared as [Sulzmann and Fürnkranz, 2009].

This in mind, we can define some statistical values of a data set which are needed for later definitions. A data set consists of $|C|$ classes and n instances from which n^c belong to the class c respectively ($n = \sum_{c=1}^{|C|} n^c$). A rule r covers n_r instances which are distributed over the classes, so that n_r^c instances belong to class c ($n_r = \sum_{c=1}^{|C|} n_r^c$).

A probabilistic rule r is an extension of a classification rule, which does not only predict a single class value, but a set of *class probabilities*, which form a probability distribution over the classes. This probability distribution estimates all probabilities that a covered instance belongs to any of the class in the data set, so we get one class probability per class. The example is then classified with the most probable class. The probability that an instance x covered by rule r belongs to c can be viewed as a conditional probability $\Pr(c|r \supseteq x)$. Thus the set of class probabilities can be noted as vector of probabilities sorted by the class ordering:

$$\vec{\Pr}(r \supseteq x) = (\Pr(c_1|r \supseteq x), \cdots, \Pr(c_{|C|}|r \supseteq x)) \qquad (2)$$

On the vector $\vec{\Pr}(r \supseteq x)$, abbreviated $\vec{\Pr}_r(x)$, we define the following maximum function

$$\max\left(\vec{\Pr}_r(x)\right) = \max_{c \in C} \Pr(c|r \supseteq x). \qquad (3)$$

On sets of class probability vectors $\bigcup_{j=1}^{k} \vec{\Pr}_i(x)$ we define the average function

$$\operatorname{avg}\left(\bigcup_{j=1}^{k} \vec{\Pr}_j(x)\right) = \frac{1}{k} \sum_{j=0}^{k} \vec{\Pr}_j(x) \qquad (4)$$

and the multiplication

$$\operatorname{mult}\left(\bigcup_{j=1}^{k} \vec{\Pr}_j(x)\right) = \left(\prod_{j=0}^{k} \Pr(c_1|r_j \supseteq x), \cdots, \prod_{j=0}^{k} \Pr(c_{|C|}|r_j \supseteq x)\right) \qquad (5)$$

Obviously the results of these functions are also class probability vectors.

In the next section, we discuss some approaches for estimating these class probabilities.

2.2 Basic Probability Estimation

In this subsection we will review three basic methods for probability estimation. Subsequently, in Section 2.3, we will describe a technique known as shrinkage, which is known from various application areas, and show how this technique can be adapted to probabilistic rule learning.

All of the three basic methods we employed, calculate the relation between the number of instances covered by the rule n_r and the number of instances covered by the rule but also belong to a specific class n_r^c. The differences between the methods are the minor modifications of the calculation of this relation.

The simplest approach to rule probability estimation directly estimates a class probability distribution of a rule with the fraction of examples that belong to each class.

$$\underset{\text{naïve}}{\Pr}(c|r \supseteq x) = \frac{n_r^c}{n_r} \qquad (6)$$

This naïve approach has several well-known disadvantages, most notably that rules with a low coverage may be lead to extreme probability values. For this reason, the use of

the Laplace- and m-estimates was suggested in [Cestnik, 1990].

The Laplace estimate modifies the above-mentioned relation by adding one additional instance to the counts n_r^c for each class c. Hence the number of covered instances n_r is increased by the number of classes $|C|$.

$$\Pr_{\text{Laplace}}(c|r \supseteq x) = \frac{n_r^c + 1}{n_r + |C|} \tag{7}$$

It may be viewed as a trade-off between $\Pr_{\text{naïve}}(c|r \supseteq x)$ and an *a priori* probability of $\Pr(c) = 1/|C|$ for each class. Thus, it implicitly assumes a uniform class distribution.

The m-estimate generalizes this idea by making the dependency on the prior class distribution explicit, and introducing a parameter m, which allows to trade off the influence of the *a priori* probability and$\Pr_{\text{naïve}}$.

$$\Pr_{m}(c|r \supseteq x) = \frac{n_r^c + m \cdot \Pr(c)}{n_r + m} \tag{8}$$

The m-parameter may be interpreted as a number of examples that are distributed according to the prior probability, which are added to the class frequencies n_r^c. The prior probability is typically estimated from the data using $\Pr(c) = n^c/n$ (but one could, e.g., also use the above-mentioned Laplace-correction if the class distribution is very skewed). Obviously, the Laplace-estimate is a special case of the m-estimate with $m = |C|$ and $\Pr(c) = 1/|C|$.

2.3 Shrinkage

Shrinkage is a general framework for smoothing probabilities, which has been successfully applied in various research areas.[2]Its key idea is to "shrink" probability estimates towards the estimates of its generalized rules r_k, which cover more examples. This is quite similar to the idea of the Laplace- and m-estimates, with two main differences: First, the shrinkage happens not only with respect to the prior probability (which would correspond to a rule covering all examples) but interpolates between several different generalizations, and second the weights for the trade-off are not specified *a priori* (as with the m-parameter in the m-estimate) but estimated from the data.

In general, shrinkage estimates the probability $\Pr(c|r \supseteq x)$ as follows:

$$\Pr_{Shrink}(c|r \supseteq x) = \sum_{k=0}^{|r|} w_c^k \Pr(c|r_k) \tag{9}$$

where w_c^k are weights that interpolate between the probability estimates of the generalized rules r_k. In our implementation, we use only generalizations of a rule that can be obtained by deleting a final sequence of conditions. Thus, for a rule with length $|r|$, we obtain $|r| + 1$ generalizations r_k, where r_0 is the rule covering all examples, and $r_{|r|} = r$.

The weights w_c^k can be estimated in various ways. We employ a shrinkage method proposed in [Wang and Zhang, 2006] which is intended for decision tree learning but can be straight-forwardly adapted to rule learning. The authors propose to estimate the weights w_c^k with an iterative procedure which averages the probabilities obtained by removing training examples covered by this rule. In effect, we obtain two probabilities per rule generalization and class: the removal of an example of class c leads to a decreased probability $\Pr_-(c|r_k \supseteq x)$, whereas the removal of an example of a different class results in an increased probability $\Pr_+(c|r_k \supseteq x)$. Weighting these probabilities with the relative occurrence of training examples belonging to this class we obtain a smoothed probability

$$\Pr_{Smoothed}(c|r_k \supseteq x) = \frac{n_r^c}{n_r} \cdot \Pr_-(c|r_k \supseteq x) \tag{10}$$

$$+ \frac{n_r - n_r^c}{n_r} \cdot \Pr_+(c|r_k \supseteq x) \tag{11}$$

Using these smoothed probabilities, this shrinkage method computes the weights of these nodes in linear time (linear in the number of covered instances) by normalizing the smoothed probabilities separately for each class.

$$w_c^k = \frac{\Pr_{Smoothed}(c|r_k \supseteq x)}{\sum_{i=0}^{|r|} \Pr_{Smoothed}(c|r_i \supseteq x)} \tag{12}$$

Multiplying the weights with their corresponding probability we obtain "shrinked" class probabilities for the instance.

Note that all instances which are classified by the same rule receive the same probability distribution. Therefore the probability distribution of each rule can be calculated in advance.

3 Rule Learning Algorithm

For the rule generation we employed the rule learner Ripper [Cohen, 1995], arguably one of the most accurate rule learning algorithms today. We used Ripper both in ordered and in unordered mode:

Ordered Mode: In ordered mode, Ripper learns rules for each class, where the classes are ordered according to ascending class frequencies. For learning the rules of class c_i, examples of all classes c_j with $j > i$ are used as negative examples. No rules are learned for the last and most frequent class, but a rule that implies this class is added as the default rule. At classification time, these rules are meant to be used as a decision list, i.e., the first rule that fires is used for prediction.

Unordered Mode: In unordered mode, Ripper uses a one-against-all strategy for learning a rule set, i.e., one set of rules is learned for each class c_i, using all examples of classes $c_j, j \neq i$ as negative examples. At prediction time, all rules that cover an example are considered and the rule with the maximum probability estimate is used for classifying the example. If no rule covers the example, it classified by the default rule predicting the majority class.

We used JRip, the Weka [Witten and Frank, 2005] implementation of Ripper. Contrary to William Cohen's original implementation, this re-implementation does not support the unordered mode, so we had to add a re-implementation of that mode.[3] We also added a few other minor modifications which were needed for the probability estimation, e.g. the collection of statistical counts of the sub rules.

In addition, Ripper (and JRip) can turn the incremental reduced error pruning technique [Fürnkranz and Widmer, 1994; Fürnkranz, 1997] on and off. Note, however,

[2]Shrinkage is, e.g., regularly used in statistical language processing [Chen and Goodman, 1998; Manning and Schütze, 1999]

[3]Weka supports a general one-against-all procedure that can also be combined with JRip, but we could not use this because it did not allow us to directly access the rule probabilities.

that with turned off pruning, Ripper still performs prepruning using a minimum description length heuristic [Cohen, 1995]. We use Ripper with and without pruning and in ordered and unordered mode to generate four set of rules. For each rule set, we employ several different class probability estimation techniques.

In the test phase, all covering rules are selected for a given test instance. Using this reduced rule set we determine the most probable rule. For this purpose we select the most probable class of each rule and use this class value as the prediction for the given test instance and the class probability for comparison. Ties are solved by predicting the least represented class. If no covering rules exist the class probability distribution of the default rule is used.

4 Experimental Results

4.1 Experimental Setup

We performed our experiments within the WEKA framework [Witten and Frank, 2005]. We tried each of the four configuration of Ripper (unordered/ordered and pruning/no pruning) with 5 different probability estimation techniques, Naïve (labeled as Precision), Laplace, and m-estimate with $m \in \{2, 5, 10\}$, both used as a stand-alone probability estimate (abbreviated with B) or in combination with shrinkage (abbreviated with S). As a baseline, we also included the performance of pruned or unpruned standard JRip accordingly. Our unordered implementation of JRip using Laplace stand-alone for the probability estimation is comparable to the unordered version of Ripper (Cohen, 1995), which is not implemented in JRip.

We evaluated these methods on 33 data sets of the UCI repository [Asuncion and Newman, 2007] which differ in the number of attributes (and their categories), classes and training instances. As a performance measure, we used the weighted area under the ROC curve (AUC), as used for probabilistic decision trees in [Provost and Domingos, 2003]. Its key idea is to extend the binary AUC to the multiclass case by computing a weighted average the AUCs of the one-against-all problems N_c, where each class c is paired with all other classes:

$$\mathrm{AUC}(N) = \sum_{c \in C} \frac{n_c}{|N|} \mathrm{AUC}(N_c) \tag{13}$$

For the evaluation of the results we used the Friedman test with a post-hoc Nemenyi test as proposed in [Demsar, 2006]. The significance level was set to 5% for both tests. We only discuss summarized results here, detailed result tables can be found in [Sulzmann and Fürnkranz, 2009] and [Sulzmann and Fürnkranz, 2010].

4.2 Ordered Rule Sets

In the first two test series, we investigated the ordered approach using the standard JRip approach for the rule generation, both with and without pruning. The basic probability methods were used standalone (B) or in combination with shrinkage (S).

The Friedman test showed that in both test series, the employed combinations of probability estimation techniques showed significant differences. Considering the CD chart of the first test series (Figure 1(a)), one can identify three groups of equivalent techniques. Notable is that the two best techniques, the m-Estimate used stand-alone with $m = 2$ and $m = 5$ respectively, belong only to the best group. These two are the only methods that are significantly better than the two worst methods, Precision used stand-alone and Laplace combined with shrinkage. On the other hand, the naïve approach seems to be a bad choice as both techniques employing it rank in the lower half. However our benchmark JRip is positioned in the lower third, which means that the probability estimation techniques clearly improve over the default decision list approach implemented in JRip.

Comparing the stand-alone techniques with those employing shrinkage one can see that shrinkage is outperformed by their stand-alone counterparts. Only Precision is an exception as shrinkage yields increased performance in this case. In the end shrinkage is not a good choice for this scenario.

The CD-chart for ordered rule sets with pruning (Figure 1(b)) features four groups of equivalent techniques. Notable are the best and the worst group which overlap only in two techniques, Laplace and Precision used stand-alone. The first group consists of all stand-alone methods and JRip which dominates the group strongly covering no shrinkage method. The last group consists of all shrinkage methods and the overlapping methods Laplace and Precision used stand-alone. As all stand-alone methods rank before the shrinkage methods, one can conclude that they outperform the shrinkage methods in this scenario as well. Ripper performs best in this scenario, but the difference to the stand-alone methods is not significant.

4.3 Unordered Rule Sets

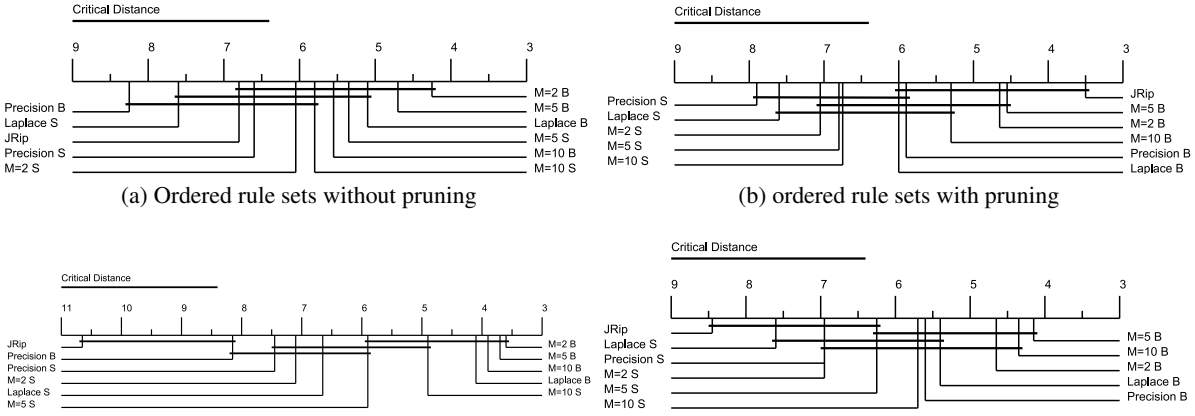
Test series three and four used the unordered approach employing the modified JRip which generates rules for each class. Analogous to the previous test series the basic methods are used as stand-alone methods or in combination with shrinkage (left and right column respectively). Test series three used no pruning while test series four did so. The results of the Friedman test showed that the techniques of test series three and test series four differ significantly.

Regarding the CD chart of test series three (Figure 1(c)), we can identify four groups of equivalent methods. The first group consists of all stand-alone techniques, except for Precision, and the m-estimates techniques combined with shrinkage and $m = 5$ and $m = 10$, respectively. Whereas the stand-alone methods dominate this group, $m = 2$ being the best representative. Apparently these methods are the best choices for this scenario. The second and third consist mostly of techniques employing shrinkage and overlap with the worst group in only one technique. However our benchmark JRip belongs to the worst group being the worst choice of this scenario. Additionally the shrinkage methods are outperformed by their stand-alone counterparts.

The CD chart of test series four (Figure 1(d)) shows similar results. Again four groups of equivalent techniques groups can be identified. The first group consists of all stand-alone methods and the m-estimates using shrinkage and $m = 5$ and $m = 10$ respectively. This group is dominated by the m-estimates used stand-alone with $m = 2$, $m = 5$ or $m = 10$. The shrinkage methods are distributed over the other groups, again occupying the lower half of the ranking. Our benchmark JRip is the worst method of this scenario.

4.4 Unpruned vs. Pruned Rule Sets

Rule pruning had mixed results, which are briefly summarized in Table 1. On the one hand, it improved the results of the ordered approach, on the other hand it worsened the results of the unordered approach. In any case, in our ex-



(a) Ordered rule sets without pruning

(b) ordered rule sets with pruning

(c) Unordered rule sets without pruning

(d) Unordered rule sets with pruning

Figure 1: Critical distance charts of ordered rule sets ((a) and (b)) and unordered rule sets ((c) and (d))

periments, contrary to previous results on PETs, rule pruning was not always a bad choice. The explanation for this result is that in rule learning, contrary to decision tree learning, new examples are not necessarily covered by one of the learned rules. The more specific rules become, the higher is the chance that new examples are not covered by any of the rules and have to be classified with a default rule. As these examples will all get the same default probability, this is a bad strategy for probability estimation. Note, however, that JRip without pruning, as used in our experiments, still performs an MDL-based form of pre-pruning. We have not yet tested a rule learner that performs no pruning at all, but, because of the above deliberations, we do not expect that this would change the results with respect to pruning.

5 Probability Estimates from Rule Ensembles

Instead of trying to improve the probability estimates for each individual leaf, one can also resort to averaging multiple estimates, thereby reducing the variance of the resulting probability estimates. For example, a technique based on bagging multiple unpruned decision trees was used in [Domingos, 1999] to obtain improved probability estimates, which were subsequently used for cost-sensitive classification. An adaptation of this technique to rule learning again has the effect that an example may be covered by a varying number of rules, whereas in decision tree learning, each example will be covered by exactly s leafs (where s is the number of trees learned).

For investigating the performance ensemble-based probability estimates, we combined our rule learner with bagging [Breiman, 1996]. We generated s bootstrap samples $S_1, \cdots, S_s$ by repeatedly drawing n instances with replacement. The rule algorithm described in Section 3 was applied to each sampled data set s_i obtaining s classifiers $C_1, \cdots, C_s$ and their corresponding rule sets $R_1, \cdots, R_s$. The probability estimation for each is computed using the previously introduced basic estimation methods.

For prediction, we determine the covering rules of each sampled rule set R_i for a given example x

$$R_i(x) = \{r \in R_i | r \supseteq x\} \tag{14}$$

Let $Cov_i(x)$ denote the set of all class probability distributions that originate from a rule in the set of covering rules

$$Cov_i(x) = \left\{ \vec{\Pr}_r(x) | r \in R_i(x) \right\}. \tag{15}$$

From this set of class probability distributions of the covering rules, we try to estimate a class probability distribution for the given example x. For this purpose we have to decode the probability estimations of these covering rules $\vec{\Pr}_r(x)$ into a single normalized global class probability distribution $\vec{\Pr}_{global}(x)$.

For our approach we considered four decoding methods. The first three methods have in common that they average the class probability distribution of (some of) the covering rules.

Best Rule: Only the most confident covering rule $\vec{\Pr}_i(x)$ of each covering rule set $R_i(x)$ is determined.

$$\vec{\Pr}_i(x) = \underset{\vec{\Pr}(x) \in Cov_i}{\arg\max} \left(\vec{\Pr}(x) \right) \tag{16}$$

Afterwards the class probability distributions of these rules are averaged and the result is normalized:

$$\vec{\Pr}_{global}(x) = \frac{\text{avg}\left(\vec{\Pr}_1, \cdots, \vec{\Pr}_s\right)}{\left\| \text{avg}\left(\vec{\Pr}_1, \cdots, \vec{\Pr}_s\right) \right\|_1}. \tag{17}$$

Macro Averaging: All covering rules are determined and their class probability distributions are macro-averaged in two steps. First the class probability distributions $Cov_i(x)$ of each covering rule set $R_i(x)$ are averaged and normalized:

$$\vec{\Pr}_i = \frac{\text{avg}\left(Cov_i(x)\right)}{\left\| \text{avg}\left(Cov_i(x)\right) \right\|_1}. \tag{18}$$

These local class probabilities are then averaged as above (17).

Micro Averaging: All covering rules are determined and their class probability distributions are micro-averaged. Essentially, this means that all learned rules are pooled, and the average is formed over the resulting set set of rules:

$$\vec{\Pr}_{global}(x) = \frac{\text{avg}\left(Cov_1(x) \cup \cdots \cup Cov_s(x)\right)}{\left\| \text{avg}\left(Cov_1(x) \cup \cdots \cup Cov_s(x)\right) \right\|_1} \tag{19}$$

Bayesian Decoding: All covering rules are pooled as above, but their class probability distributions are multiplied with each other and with the vector of the a priori class probabilities

$$\vec{\Pr}_{prior} = \left(\Pr(c_1), \cdots, \Pr(c_{|C|})\right).$$

Table 1: Unpruned vs. pruned rule sets: Win/Loss for ordered (top) and unordered (bottom) rule sets

	JRip	Precision		Laplace		M 2		M 5		M 10	
Win	26	23	19	20	19	18	20	19	20	19	20
Loss	7	10	14	13	14	15	13	14	13	14	13
Win	26	21	9	8	8	8	8	8	8	8	6
Loss	7	12	24	25	25	25	25	25	25	25	27

Thus $\vec{\Pr}_{global}(x)$ is calculated as follows

$$\vec{\Pr}_{global}(x) = \frac{\text{mult}\left(\bigcup_{i=1}^{s} Cov_i(x) \cup \left\{\vec{\Pr}_{prior}\right\}\right)}{\left\|\text{mult}\left(\bigcup_{i=1}^{s} Cov_i(x) \cup \left\{\vec{\Pr}_{prior}\right\}\right)\right\|_1} \tag{20}$$

In all cases, the resulting class probability distribution $\vec{\Pr}_{global}(x)$ is used for the prediction. For this purpose the most probable class according to $\vec{\Pr}_{global}(x)$ is selected and this class value is used as the prediction for the given test instance x. Ties are solved by predicting the least represented class. If no covering rules ($Cov_i(x) = \emptyset$) exist for a sampled rule set R_i the class probability distribution of the default rule is used accordingly.

6 Results on Ensemble-Based Probability Estimates

6.1 Experimental Setup

The above methods were integrated into the framework described in Section 4.1. For sampling, we employed the unsupervised random sampling of WEKA for the generation of the bootstrap samples and applied the Bagging implementation of WEKA to JRip. In accordance with the results obtained above, we only used unordered rule sets for these experiments because these produce better probability estimates than ordered rule sets. Furthermore we know that the unpruned rule sets perform better for the Laplace- and m-estimates than pruned rule sets if the rule sets are generated by the unordered JRip. For Precision the opposite is true. So we employed these basic probability estimation techniques on either unpruned or pruned rule sets according to these observations. As the employed shrinkage method worsened the probability estimation we abstained from using shrinkage in these experiments. All previously mentioned decoding methods - Best rule, Macro and Micro Averaging, and Bayesian Decoding - were employed. So we computed all combinations of the basic probability estimation and decoding methods on a different number of bootstrap samples - 10, 20, 50 and 100 samples accordingly. These methods were compared to the default configuration of JRip (pruned, ordered rule sets) and to its bagged version (with or without pruning) using the same bootstrap samples.

All methods were evaluated on the 33 previously used data sets of the UCI repository. As a performance measure, we used the weighted area under the ROC curve (AUC) also. For our comparison we calculated the average weighted AUC over all data sets for all combinations (combining a probability estimation technique and a decoding method to a given number of boostrap samples), the summarized results are depicted in Figure 2. Detailed results can be found in [Sulzmann and Fürnkranz, 2010].

6.2 Comparison to JRip

In our experiments we compared JRip and its bagged versions to the basic probability estimation techniques employing the four decoding methods. Figure 2 shows the results of unpruned bagged JRip and the basic probability estimation techniques. We omitted to depict the results of JRip and the pruned bagged JRip because they both had a worse performance than the unpruned bagged version which has the worst performance of the depicted methods. The weighted AUC of their best representative, unpruned bagged JRip, was always at least two absolute percentage points lower than the weighted AUC of the basic probability estimation techniques for all decoding methods and all numbers of samples. So we can conclude that the performance of the basic probability estimation techniques improve over the default probability estimation integrated in JRip. As this observation was also made in the basic rule learning experiments, we see our approach reconfirmed.

6.3 Comparison of the base probability estimation techniques

Each of the results of Figure 2 is the average performance over several different base probability estimation techniques: Precision applied to pruned rule sets and the Laplace and the m-estimate ($m \in \{2, 5, 10\}$) applied to unpruned rule sets. The applied decoding methods seem to have only a small impact on the ranking (according the weighted AUC) of the basic probability estimation techniques. Especially for a higher number of bootstrap samples, 50 and 100, these rankings are always the same - Precision having the best performance followed by the m-estimate using $m = 2$, $m = 5$, or $m = 10$ and the Laplace estimate in this order. For the smaller numbers of samples, 10 and 20, the ranking is a little bit more dynamic. Although the best two methods, m-estimate with $m = 2$ and $m = 5$ (in this order), and the worst method, Laplace estimate, are always the same, the methods in the center, Precision and the m-estimate using $m = 10$, switch places dependent on the decoding methods. So we can conclude that the m-estimate using $m = 2$ is the best choice for a low number of samples just as Precision is the best choice for a higher number of samples. For all probability estimation techniques holds that an increase in the number of samples leads to an improvement in the weighted AUC but the gain is a bit lower than the gain of the bagged JRip.

6.4 Comparison of the decoding methods

In this section, we want to compare the four employed decoding methods: Best Rule, Macro and Micro Averaging, and Bayesian Decoding. For this purpose we calculated the average weighted AUC over all data sets for all combinations obtained by combining a probability estimation technique, a decoding method and a number of bootstrap samples. Afterwards, we determined on this data the average rank of each decoding method (Table 2) and used this information for a Friedman test with a significance of 5%. According to this test, the decoding methods differ significantly, so we applied a post-hoc Nemenyi test which is depicted in a CD chart (Figure 3).

Table 2: Decoding Methods: Count of each rank and the average rank

Decoding Method	Ranked 1st	2nd	3rd	4th	Average Rank
Best Rule	0	0	4	16	3.80
Macro Averaging	0	17	3	0	2.15
Micro Averaging	20	0	0	0	1.00
Bayesian Decoding	0	3	13	4	3.05

Obviously, Micro Averaging is the best decoding methods in our experiments as it always placed first according to the average weighted AUC. This observation is reconfirmed by the CD chart since the Micro Averaging is the only member of the best group of methods. Macro Averaging and Bayesian Decoding do not differ significantly being both in the second best group of decoding methods. Nevertheless Bayesian Decoding is also in the worst group together with Best Rule which is the worst choice in our experiments.

The observed ranking of the decoding methods can be attributed to the individual exploitation of the covering rules. As the decoding method Best Rule only uses the best rule of each bootstrap sample, a great deal of evidence has no influence on the probability estimation. Thus, it is not surprising that Best Rule ranks behind the methods that make better use of the ensemble.

The methods Macro and Micro Averaging both average the probability distributions of a number of covering rules but their averaging approaches differ. So the influence of a high evidence for a class in a sampled rule set is also different for the two methods. For Micro Averaging the aforementioned evidence has a direct effect on resulting probability distribution as all covering rules have the same weight in its calculation. As our rule sets only have a low redundancy, this effect is desirable. For Macro Averaging a high evidence in a sampled rule set influences only the probability distribution of this bootstrap sample. So the number of covering rules has no effect on the global probability distribution. As Macro Averaging partially discards the available information Micro Averaging should perform better than Macro Averaging as observed in our experiments.

Bayesian Decoding uses all the information contained in the covering rules as Micro Averaging does. These two methods differ only the way how they combine the information of the covering rule, averaging or multiplying their probability distributions. The multiplication used in the Bayesian approach has a tendency to prefer a number of medium probabilities to a balanced number of low and high probabilities. This bias has a negative effect on the calculation of the global probability distribution. Averaging is more desirable as high probabilities have a greater impact on its calculation.

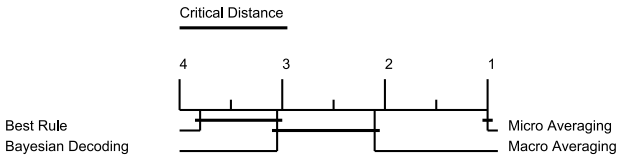


Figure 3: CD chart of the decoding methods

7 Conclusions

The most important result of our study is that probability estimation is clearly an important part of a good rule learning algorithm. The probabilities of rules induced by JRip can be improved considerably by simple estimation techniques. In unordered mode, where one rule is generated for each class, JRip is outperformed in every scenario. On the other hand, in the ordered setting, which essentially learns decision lists by learning subsequent rules in the context of previous rules, the results were less convincing, giving a clear indication that the unordered rule induction mode should be preferred when a probabilistic classfication is desirable.

Among the tested probability estimation techniques, the m-estimate typically outperformed the other methods in our version of the JRip algorithm. The superiority of the m-estimate was not sensitive to the choice of its parameter. When combined with an ensemble-based approach, the m-estimate maintained its superiority for smaller number of bootstrap samples, but typically lower value of m performed better. For higher numbers of bootstrap samples, precision, which corresponds to a value of $m = 0$, outperformed the other methods. Thus, it seems to be the case that the use of the m-estimate primarily helps to reduce the variance of the probability estimates.

The employed shrinkage method did, in general, not improve the simple estimation techniques. It remains to be seen whether alternative ways of setting the weights could yield superior results. Rule pruning did not produce the bad results that are known from ranking with pruned decision trees, presumably because unpruned, overly specific rules will increase the number of uncovered examples, which in turn leads to bad ranking of these examples.

Ensemble-based probability estimation based on a bagging approach further improved the probability estimates. The improvement increases with the number of bootstrap samples. In every case, the probabilistic approach outperformed Bagged JRip using the same number of bootstrap samples. Amongst the employed decoding methods, Micro Averaging of the probability distributions of all covering rules was without exceptions superior to the other methods, indicating that the predictions of rule-based ensembles should be combined at the level of individual rules and not at the level of theories.

Acknowledgements

This research was supported by the *German Science Foundation (DFG)* under grant FU 580/2.

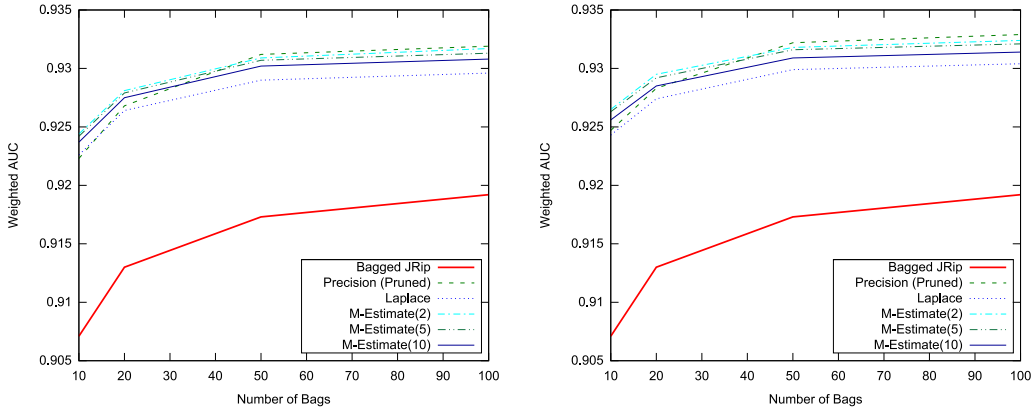
References

A. Asuncion and D. J. Newman. UCI machine learning repository, 2007.

Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

Bojan Cestnik. Estimating probabilities: A crucial task in Machine Learning. In L. Aiello, editor, *Proceedings of the 9th European Conference on Artificial Intelligence (ECAI-90)*, pages 147–150, Stockholm, Sweden, 1990. Pitman.

Stanley F. Chen and Joshua T. Goodman. An empirical study of smoothing techniques for language modeling. Technical Report TR-10-98, Computer Science Group, Harvard University, Cambridge, MA, 1998.



(a) Best Rule

(b) Macro Averaging

(c) Micro Averaging

(d) Bayesian Decoding

Figure 2: Average Weighted AUC of the employed decoding methods on unpruned rule sets (where stated the pruned rule set is used)

William W. Cohen. Fast effective rule induction. In A. Prieditis and S. Russell, editors, *Proceedings of the 12th International Conference on Machine Learning (ML-95)*, pages 115–123, Lake Tahoe, CA, 1995. Morgan Kaufmann.

Janez Demsar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006.

Pedro Domingos. Metacost: A general method for making classifiers cost-sensitive. In *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-99)*, pages 155–164, San Diego, CA, 1999. ACM.

César Ferri, Peter A. Flach, and José Hernández-Orallo. Improving the AUC of probabilistic estimation trees. In Nada Lavrac, Dragan Gamberger, Ljupco Todorovski, and Hendrik Blockeel, editors, *Proceedings of the 14th European Conference on Machine Learning*, pages 121–132, Cavtat-Dubrovnik, Croatia, 2003. Springer.

Johannes Fürnkranz and Gerhard Widmer. Incremental Reduced Error Pruning. In William W. Cohen and H. Hirsh, editors, *Proceedings of the 11th International Conference on Machine Learning (ML-94)*, pages 70–77, New Brunswick, NJ, 1994. Morgan Kaufmann.

Johannes Fürnkranz. Pruning algorithms for rule learning. *Machine Learning*, 27(2):139–171, 1997.

Eyke Hüllermeier and Stijn Vanderlooy. Why fuzzy decision trees are good rankers. *IEEE Transactions on Fuzzy Systems*, 17(6):1233–1244, 2009.

Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, MA, 1999.

Foster J. Provost and Pedro Domingos. Tree induction for probability-based ranking. *Machine Learning*, 52(3):199–215, 2003.

Jan-Nikolas Sulzmann and Johannes Fürnkranz. An empirical comparison of probability estimation techniques for probabilistic rules. In João Gama, Vítor Santos Costa, A. Jorge, and Pavel B. Brazdil, editors, *Proceedings of the 12th International Conference on Discovery Science (DS-09)*, pages 317–331. Springer-Verlag, 2009. Winner of Best Student Paper Award.

Jan-Nikolas Sulzmann and Johannes Fürnkranz. Probability estimation and aggregation for rule learning. Technical Report TUD-KE-2010-03, TU Darmstadt, Knowledge Engineering Group, 2010.

Bin Wang and Harry Zhang. Improving the ranking performance of decision trees. In Johannes Fürnkranz, T. Scheffer, and M. Spiliopoulou, editors, *Proceedings of the 17th European Conference on Machine Learning (ECML-06)*, pages 461–472, Berlin, Germany, 2006. Springer-Verlag.

Ian H. Witten and Eibe Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco, CA, 2nd edition, 2005.

Conditional Random Fields For Local Adaptive Reference Extraction

Martin Toepfer and **Peter Kluegl** and **Andreas Hotho** and **Frank Puppe**
University of Würzburg,
Department of Computer Science VI
Am Hubland, 97074 Würzburg, Germany
{toepfer, pkluegl, hotho, puppe}@informatik.uni-wuerzburg.de

Abstract

The accurate extraction of bibliographic information from scientific publications is an active field of research. Machine learning, especially sequence labeling approaches like Conditional Random Fields (CRF), are often applied for this reference extraction task, but still suffer from the ambiguity of reference notation. Reference sections apply a predefined style guide and contain only homogeneous references. Therefore, other references of the same paper or journal often can provide evidence how the fields of a reference are correctly labeled. We propose a novel approach that exploits the similarities within a document. Our process model uses information of unlabeled documents directly during the extraction task in order to automatically adapt to the perceived style guide. This is implemented by changing the manifestation of the features for the applied CRF. The experimental results show considerable improvements compared to the common approach. We achieve an average F_1 score of 96.7% and an instance accuracy of 85.4% on the test data set.

1 Introduction

Reference sections of research papers are a valuable source for many interesting applications. A considerable amount of research has been spent on creating and analyzing citation graphs, yielding information about research communities and topics. Social bookmarking services like Bibsonomy[1] on the other hand have become essential tools for researchers and facilitate the management of bibliographic data. Both applications, citation analysis and bookmarking services, rely on a structured representation of the reference data. Often the well-known BibTex format is used to define the different fields of an information. The acquisition of this structured data demands for an automatic processing of the vast amount of the unstructured data available in publications.

The knowledge for an automatic extraction of references can be formalized using rules or templates. However, the handcrafting of rules is tedious and prone to error due to the knowledge engineering bottleneck. Several publications have shown that machine learning and especially sequence labeling approaches are more suitable for the reference extraction task [Peng and McCallum, 2004; Councill *et al.*, 2008]. These methods learn a statistical model using training sets where the interesting information, namely the BibTeX fields, is already labeled. The model is applied on newly and unseen documents in order to identify the information in unlabeled data. Hence, the model is only adapted offline on the previously seen documents of the training phase. Although these approaches achieve remarkable results, the heterogeneous styles of the references make a suitable generalization difficult and decrease the accuracy of the extraction task. The IEEE style, for example, separates the author and the title with a comma and surrounds the title with quotes. Whereas the ACM style applies no separator for the author and the date is located between the author and the title. The MISQ style surrounds the title also with quotes, but uses no separator for the author. Nevertheless, the input data of the extraction task, i.e., the reference section of scientific publications, follows a single style guide. The references within a paper or journal are usually homogenous. In order to utilize these local consistencies the model has to be adapted during the extraction phase, because the applied style guide is identified as the document is processed. This is not possible using the common process model.

In this paper, we propose a local adaptive information extraction approach using sequence labeling methods, especially Conditional Random Fields. That is a novel extension of the common process model with an automatic adaption to the previously unknown style guide. We apply two stacked models that are trained offline. The first model is applied to gain information about the document's structure and to create a description of the reference notation based on the available features. The style guides differ in their characteristics of field separation and alignment. As a result, the description, also called local model, is based on different features dependent on the currently processed document. This information is used to create style-specific features which have a steady meaning for the information extraction task. However, the manifestation of these meta features differs between documents and depends on the applied style guide. The new features are then added to the features of the second model helping to resolve ambiguities and to increase the extraction accuracy. As a result, the presented approach achieves considerably better results than a single Conditional Random Field.

The rest of the paper is structured as follows: Section 2 introduces the novel combination of methods and gives a detailed description of all parts of the process model. Then, the evaluation setting and experimental results are presented and discussed in section 3. Section 4 gives a short overview of the related work and section 5 concludes with a summary of the presented work.

[1] http://www.bibsonomy.org/

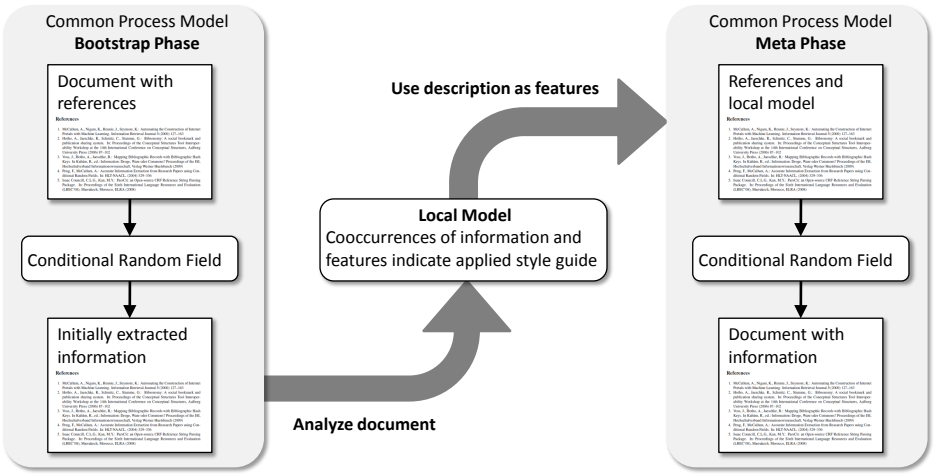


Figure 1: Overview of the applied process model with three phases: the bootstrap, local model and meta phase.

2 Method

Machine learning and sequence labeling approaches are often applied for reference extraction and Conditional Random Fields (CRF) are one of the most popular techniques for this task. Normally, a simple process model is used: The feature extraction identifies valuable properties in the unstructured data. They are used by a given model in order to extract interesting information, that is, labeling the fields of a reference. The extraction model is trained on labeled examples in a previous phase. Therefore, the model is only adapted offline in the learning process on the global consistencies of the domain. This prevents a good generalization for the global model and induces errors in the extracted information. In order to overcome this problem, the local patterns and the consistency of one document, the applied style guide in this domain respectively, need to be addressed directly for a resolution of the ambiguity. The style guide can however be identified as soon as the document is processed. Hence, the common process model is incompatible to an online adaption during the extraction process on the local consistency.

The presented approach tries to utilize the common process model with CRFs in a novel combination. The unlabeled documents are used to identify the applied style guide directly during the extraction process. Then, this information about the homogenous notation within the current document is exploited to increase the extraction accuracy in an additional phase. The model is learnt offline, but the features it is based on are adapted online during the extraction process of each single document. Since the process can adjust to the local consistencies, it is called *local adaptive*. Figure 1 provides an overview of the applied process that consists of three stages: the bootstrap, the construction of the local model and the meta phase. The purpose of the *bootstrap phase* is to provide the fundamental information that is needed to perceive style information for a document. We employ a common process model as it can be found in previous CRF approaches. Features are extracted from the current document and a previously learnt (base) model is applied in order to gain information. However, this is just an intermediate step required for an examination of the local information patterns. The second phase, the construction of the *local model*, tries to create a description of the applied style guide. This is achieved by investigating the cooccurrence of information and features and the selection of features that describe the different characteristics of the style guide very well. Finally, the acquired style information is used to create special features, called meta features. These possess a different manifestation for each document. The *meta phase* is the final step of the process model. It is built on the common process model of conditional random fields, but uses an enhanced set of features. Additionally to the base features of the bootstrap, it also considers the new meta features that provide hints on how the information is structured in the applied style guide.

In summary, the bootstrap phase takes an initial look at the reference section and handles apparent style information over to the meta phase which finally processes the reference section as if the applied style guide was known.

For a detailed description of the process, first the applied terminology is presented. Then, conditional random fields and the usage of features are addressed. The elements of the local model that contain the knowledge of the document's structure build an important part of the approach and are described in detail with an example.

2.1 Terminology

In the presented approach different frameworks and toolkits are combined. In order to clarify the terminology we explain some central terms. We use a nomenclature oriented at the Apache UIMA framework [Ferrucci and Lally, 2004].

Definition 1 (Typesystem, Information Type)**.** *A* typesystem *is a set* $\mathbb{T}$*, whose elements are called* (annotation or information) types*.*

As an example, we define the label type system $\mathbb{T}_{\text{label}} = \{\text{AUTHOR, BOOKTITLE, DATE, EDITOR, INSTITUTION, JOURNAL, LOCATION, NOTE, PAGES, PUBLISHER, TECH, TITLE, VOLUME}\}$ which contains all field labels. Furthermore, we introduce the overall typesystem $\mathbb{T}_{\text{all}}$ which contains all types.

Definition 2 (Annotation)**.** *Given a text document* D *and a typesystem* $\mathbb{T}$*, we define an* annotation *as a triplet* $(s, i, j) \in \mathbb{T} \times \mathbb{N} \times \mathbb{N}$*, consisting of an information type* $s \in \mathbb{T}$ *and two naturals* $i \leq j$*, indicating the begin and the end of the annotation in* D*.*

For instance, we can assign an annotation $(\text{NUM}, 28, 32)$ to a document to state that the

text covered by the offsets 28 and 32 is a number. Therefore, we define an appropriate typesystem $\mathbb{T}_{\text{feat}} = \{\text{COMMA, CW, SW, NUM, FirstName}\dots\}$ with $\mathbb{T}_{\text{feat}} \cap \mathbb{T}_{\text{label}} = \emptyset$. The typesystem $\mathbb{T}_{\text{feat}}$ contains several useful low level information types called features, e.g., COMMA indicating commas, NUM indicating numbers, CW for capitalized words, SW for lower case words and FirstName indicating first names. These annotations are automatically assigned by the feature extraction, e.g. a word list with first names is provided and for each occurrences of an entry an annotation of the type FirstName is created.

Moreover, we partition documents into pieces of atomic lexical units, called *tokens*, to make use of the ClearTK framework [Ogren *et al.*, 2008] and the machine learning toolkit Mallet[2] for the implementation of the CRF.

Definition 3 (Token)**.** *We postulate $\tau \in \mathbb{T}_{token}$ (the Token-Type) to be a type which satisfies the following conditions.*

- *Annotations of the type τ do not cover white space characters and*
- *all other characters are covered of exactly one annotation of the type τ.*

Annotations of the type τ are called tokens.

Punctuations and special characters are put in single tokens. Alphabetic and numerical character sequences are split into separate token sets.

Definition 4 (Feature)**.** *Iff a token x_t is within*[3] *an annotation of a type $\varphi \in \mathbb{T}_{feat}$, we say that x_t has the* feature *φ.*

If a token $x_a = (\tau, 6, 7)$ has the feature COMMA $\in \mathbb{T}_{\text{feat}}$, then the text covered by the token is a comma. The terms feature and type are used synonymously in the following sections.

2.2 Conditional Random Fields

Conditional Random Fields (CRF) [Lafferty *et al.*, 2001] model conditional probabilities with undirected graphs. As usual in information extraction and sequence labeling tasks, we use linear chain CRFs. That is, we take a sequence of tokens $\mathbf{x} = (x_1, \dots, x_T)$ as input. Given binary feature functions $f_1, \dots, f_K$ and parameters $\lambda_1, \dots, \lambda_K \in \mathbb{R}$, we compute the conditional probability of the label sequence $\mathbf{y} = (y_1, \dots, y_T)$ under $\mathbf{x}$ by

$$P_\lambda(\mathbf{y}|\mathbf{x}) = \frac{1}{Z_\mathbf{x}} \exp\left(\sum_{t=1}^{T}\sum_{i=1}^{K} \lambda_i \cdot f_i(y_{t-1}, y_t, \mathbf{x}, t)\right),$$

with a normalization factor

$$Z_\mathbf{x} = \sum_{\mathbf{y}' \in Y} \exp\left(\sum_{t=1}^{T}\sum_{i=1}^{K} \lambda_i \cdot f_i(y'_{t-1}, y'_t, \mathbf{x}, t)\right).$$

Y is the set of all possible label sequences $\mathbf{y}'$ for $\mathbf{x}$.

In short, a feature function $f_i(y_{t-1}, y_t, \mathbf{x}, t))$ can testify evidence for the token at the position t to be labeled as y_t, depending on the label of it's predecessor and the observed input sequence. The feature functions are weighted by parameters $\lambda_1, \dots, \lambda_K$. Hence, if $f_i(y_{t-1}, y_t, \mathbf{x}, t)) = 1$ and λ_i has a high value, then we have strong evidence for labeling x_t as y_t. Accordingly, the parameters determine how we infer the labels from the information given by the feature functions, i.e., we assume the label sequence that is most likely given some observation sequence. As usual in supervised machine learning, we use a learning algorithm which sets the weights to make good predictions on a training set.

In principle, a feature function can make complex use of the whole input sequence, the current label and the predecessing label. However, we mainly use simpler feature functions, named *annotation-based feature functions*, which factorize into two parts. Given two labels $y_a, y_b \in \mathbb{T}_{\text{label}}$, a typesystem $\mathbb{T}$ and a type $\varphi \in \mathbb{T}_{\text{feat}}$, an annotation based feature function has the form:

$$f_{\varphi,y_a,y_b}(y_{t-1}, y_t, \mathbf{x}, t) = \mathbf{1}_{\{y_{t-1}=y_a\}} \cdot \mathbf{1}_{\{y_t=y_b\}} \cdot f_\varphi(x_t).$$

The first part is only an indication of the label transition and ensures that we can learn separate weights for each combination of labels. On the contrary, the second part is independent from the labels. $f_\varphi(x_t)$ just shows if the token at the position t has the feature φ. In different words,

$$f_\varphi(x_t) = \begin{cases} 1, & \text{if } x_t \text{ has the feature } \varphi, \\ 0, & \text{otherwise.} \end{cases}$$

Finally, we create annotation based feature functions f_{φ,y_a,y_b} for every $y_a, y_b \in \mathbb{T}_{\text{label}}$ and every type $\varphi \in \mathbb{T}_{\text{feat}}$. By example, the CRF learns a parameter $\lambda_{\text{NUM,AUTHOR,YEAR}}$ for the feature function $f_{\text{NUM,AUTHOR,YEAR}}$, i.e., a weight for having the NUM (number) feature and transitioning from an author field to a year field.

2.3 Local Adaptivity

The local model phase is the main part of the local adaptivity. It analyses the given features and the initially extracted information of the bootstrap phase and creates a description of the characteristics of the applied style guide. This description is then projected as features in order to be useful for the CRF in the meta phase. Therefore, the local model consists of two major steps: the creation of the description and the projection of features. Overall, a local model can be seen as a representation of specific knowledge of each single document's structure. There are various means to describe these local patterns of a document. The rule-based approach for local adaptivity [Kluegl *et al.*, 2010] has shown that two characteristics are describing the applied style guide sufficiently for a considerable increase of accuracy.

field separation One important consistency inside a reference section is the way how fields are separated. For instance, one writer always ends the author lists with a period, another writer may use a colon instead. If such a field separator is once determined with the help of other references, then it can help solving ambiguos cases, for example, in the case when one of the first tokens of the title also contains a colon. For every label type $\varphi_{\text{label}} \in \mathbb{T}_{\text{label}}$ we try to detect features which indicate the begin or the end of φ_{label} fields in a document. These additional features $\text{BEGIN}_{\varphi_{\text{label}}}$ and $\text{END}_{\varphi_{\text{label}}} \in \mathbb{T}_{\text{meta-feat}}$ then indicate the document specific seperators for the meta phase. For instance, if a token x_t has the feature LPAREN and we have recognized that date fields begin with a left parenthesis in this document, then we assign an annotation of the type $\text{BEGIN}_{\text{DATE}} \in \mathbb{T}_{\text{meta-feat}}$ to x_t to state that

[2] `http://mallet.cs.umass.edu`

[3] A token (τ, a, b) is within an annotation (φ, x, y), iff a $\geq$ x and b $\leq$ y.

we have evidence for the begin of a date field. In addition to these two meta features, we also introduce two specialized meta features $\text{BEGIN}^t_{\varphi_{\text{label}}}$ and $\text{END}^t_{\varphi_{\text{label}}} \in \mathbb{T}_{\text{meta-feat}}$ that restrict the projection of the feature dependent on the initally extracted information.

field sequence Style guides define not only the way of field separation. The sequence and alignment of the fields normally does not change within a reference section. Although some fields are optional and may be skipped by the author, information about the occurring sequences can resolve ambiguities and be of assistance in classification. As a simple example, we refer to the date field of the reference. Normally, the date is located either directly after the author or near the end of the reference. If no features indicate a date in the current reference, then information about the field before and the field after the dates of the remaining references hepls to find the date. For every label type $\varphi_{\text{label}} \in \mathbb{T}_{\text{label}}$ we try to detect fields that are normally located before and after the fields with the label φ_{label}. These additional features $\text{BEFORE}_{\varphi_{\text{label}}}$ and $\text{AFTER}_{\varphi_{\text{label}}} \in \mathbb{T}_{\text{meta-feat}}$ then indicate the inherent sequences of the reference section for the meta phase. For instance, if the analysis of the extracted information is confident that the field date is always followed by the pages field, then we assign an annotation of the type $\text{AFTER}_{\text{DATE}} \in \mathbb{T}_{\text{meta-feat}}$ to each token that was labeled with type φ_{PAGES}.

Summarizing, annotations of the types $t \in \mathbb{T}_{\text{meta-feat}}$ are used to enrich the feature functions of the meta phase. In the following, we describe how these types are determined with the use of the given annotation-based features and the initially extracted information.

First, an observed meta feature is created for the selection of types that are suitable for a meta feature.

Definition 5 (Observed Meta Feature)**.** *$\varphi^*_{meta} \in \mathbb{T}_{meta\text{-}observed}$ is defined as the manifestation of the corresponding meta feature $\varphi_{meta} \in \mathbb{T}_{meta\text{-}feat}$ in the case that the information was extracted perfectly.*

In other words, the observed meta feature $\varphi^*_{\text{meta}} \in \mathbb{T}_{\text{meta-observed}}$ is automatically assigned to those tokens of each reference that are located exact at the positions indicated by the meta feature $\varphi_{\text{meta}} \in \mathbb{T}_{\text{meta-feat}}$. The observed meta features $\text{BEGIN}^*_{\text{AUTHOR}}$, for example, is assigned to the first token of the initially extracted author field and the observed meta features $\text{END}^*_{\text{AUTHOR}}$ is assigned to the last token of the author field.

The observed meta features will only be utilized to determine suitable types for the meta features. All available types of features $\varphi \in \mathbb{T}_{\text{feat}}$ are compared to the observed meta features φ^*_{meta} using a similarity measure. It is useful to consider the shape of the tokens and their properties to be the outcome of a stochastic event. From this point of view, f_φ (cf. section 2.2) is a random variable and $p(f_\varphi{=}1)$ represents the probability that a token has the feature φ. Additionally, $p(f_{\varphi_1}{=}1, f_{\varphi_2}{=}1)$ is a joint probability, indicating how likely a token has both the feature φ_1 and the feature φ_2. By example, $p(f_{\text{NUM}}{=}1, f_{\text{CW}}{=}1) = 0$ since tokens cover either numbers or capitalized words.

The mutual information has shown to be a sound similarity measure. Between two random variables X and Y

$$\text{MI}(X,Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) \cdot \log \left(\frac{p(x,y)}{p_X(x) \cdot p_Y(y)} \right)$$

measures how much information X and Y share. It covers all possible outcomes of X and Y. However, since we are only interested in the coincidence of a feature φ and the observed meta features φ^*_{meta} and not, for example, in the absence of a feature, the sum of all different values or occurrences of the features is removed. Hence, the mutual information can be reduced to a weighted pointwise mutual information. The probability distributions are then estimated by the observed frequencies $\hat{p}$ in the reference section:

$$\alpha(\varphi, \varphi^*_{\text{meta}}) = \hat{p}(f_\varphi{=}1, f_{\varphi^*_{\text{meta}}}{=}1) \cdot \log \frac{\hat{p}(f_\varphi{=}1, f_{\varphi^*_{\text{meta}}}{=}1)}{\hat{p}(f_\varphi{=}1) \cdot \hat{p}(f_{\varphi^*_{\text{meta}}}{=}1)}$$

High values of $\alpha(\varphi, \varphi^*_{\text{meta}})$ indicate that the type φ is suitable to describe the meta feature φ_{meta}. The weighted pointwise mutual information is motivated with the fact the rare occurrences of an information and a feature aren't representative for the complete reference section.

After applying the formula on all available features, we gain a sorted list of rated candidates for each meta feature. For the presented work no conjunctions of features for the manifestation of a meta feature are utilized. However, a meta feature cannot always be described by a single feature, but requires sometimes a disjunction of features. Therefore, instead of only using the highest rated feature for the description of the meta feature, each feature is consulted that fulfills two conditions: its α rating exceeds a given threshold β and the annotations of the feature are disjoint to the other selected features whereas higher rated features are preferred. On the one hand, some rare applied style guides are able to create different separators for a field. But also with a strict style guide applied, the absence of some information can require a description of a meta feature with several features. If the date contains an information about the month in fifty percent of its occurrences, then the start separator of the date is either a number or a word indicating a month name. Hence, the description of the begin of the date would be described best with two features. For that reason, several feature are allowed for the manifestation, but only if they are not redundant, i.e. are disjoint to the already selected features of higher rating. For the computation of disjoint features the joint probability with the observed frequencies is reused. Two features φ_1 and φ_2 are considered disjoint iff $\hat{p}(f_{\varphi_1} = 1, f_{\varphi_2} = 1) \approx 0$. A minimal margin was applied since we assume a fallible feature extraction that erroneously assigns a feature on rare occasions.

The threshold is applied because of the weighted pointwise mutual information. Features that only occur once or twice in a document are not confident enough for the description of the local model even if they are disjoint with the already selected features. This selection of disjoint features does not need to be applied for the sequences of the fields due to the characteristics of the reference parsing domain where the labels build a disjoint partition of the complete reference. Only the threshold is used to filter rare sequences of fields.

In order to use the meta features in the common process model, the types of annotations are projected by providing an annotation-based feature function. Additionally to

the already described feature function, a specialized feature function $f_{\varphi^t_{\text{meta}}}$ is added for the separation of the fields.

$$f_{\varphi^t_{\text{meta}}}(x_t) = \begin{cases} 1, & \text{if } x_t \text{ has the feature } \varphi_{\text{meta}} \\ & \text{and } |t-o| \text{ is minimal with} \\ & f_{\varphi^*_{\text{meta}}}(x_o) = 1, \\ 0, & \text{otherwise.} \end{cases}$$

Here, a token possesses this meta feature only if it is located nearest to the observed meta feature φ^*_{meta}. The projection is of course limited to the currently considered reference. The combination of both strategies for the projection of separators enforces the reuse and simultaneously the correction of the initially extracted information.

The complete process of the creation of the local model and its projection as features is summarized in algorithm 1:

Algorithm 1 Local Model Phase

```
for all φ_meta ∈ T_meta-feat do
    l ← new list
    for all φ ∈ T_feat do
        if (∀φ' ∈ l : p̂(f_φ = 1, f_φ' = 1) ≈ 0) ∧
        α(φ, φ*_meta) > β then
            add φ to l
    project l as manifestation of φ_meta
```

2.4 Example

The selection of features and their projection as meta features are illustrated with a simplified example focusing on the meta feature $\text{END}_{\text{AUTHOR}}$. The input of the presented approach is a reference section with 20 references and overall 799 tokens that have been labeled in the bootstrap phase. Figure 2 contains two references of the reference section. The first row shows the begin of the reference, whereas the labels assigned by the bootstrap phase are depicted in the second row. Obviously, the CRF falsely labeled the tokens "Exokernel:" as author in the first reference. In the next four rows, a selection of features are added. *PM* stands for all punctuation marks, *PERIOD* for periods, *COLON* for colons and *PeriodSep* for periods that are not part of abbreviations or name initials. Finally, the last two rows contain the computed meta features $\text{END}_{\text{AUTHOR}}$ and $\text{END}^t_{\text{AUTHOR}}$. Applying the similarity measure on the given features results in a rating how good the feature describes the end of the author. The values are given for two suitable features:

$$\alpha(\varphi_{\text{PERIOD}}, \text{END}^*_{\text{AUTHOR}}) = \frac{18}{799} \cdot \log \frac{\frac{18}{799}}{\frac{143}{799} \cdot \frac{19}{799}} = 0.0375$$

$$\alpha(\varphi_{\text{PeriodSep}}, \text{END}^*_{\text{AUTHOR}}) = \frac{18}{799} \cdot \log \frac{\frac{18}{799}}{\frac{63}{799} \cdot \frac{19}{799}} = 0.0560$$

Since a token with the feature $\varphi_{\text{PeriodSep}}$ always has the feature φ_{PERIOD}, both features are not disjoint. The local model now states that the end of the author is at best described by a single feature. Hence, $\varphi_{\text{PeriodSep}}$ is assigned to $\text{END}_{\text{AUTHOR}}$ in this document and the feature function $f_{\text{END}_{\text{AUTHOR}}}(x_t) = 1$, iff x_t has the feature $\varphi_{\text{PeriodSep}}$. In a different reference section, for example, with another style guide applied $f_{\text{END}_{\text{AUTHOR}}}(x_t) = 1$, iff x_t has the feature φ_{COMMA} or φ_{COLON}. The meta feature $\text{END}^t_{\text{AUTHOR}}$ is consequently only assigned to the token that is nearest to the observed meta feature $\text{END}^*_{\text{AUTHOR}}$. That is the 20th token in the first reference and the 8th token in the second reference. The CRF of the meta phase now has access to additional features of high quality resulting in an increased accuracy.

3 Experimental Study

We have evaluated the presented process model, the idea of the local adaptivity and its novel combination of state of the art methods in an experimental study. First, the applied data sets, features and settings of the study are described. Then, the results of the evaluation are presented and discussed.

3.1 Data sets

The labeled data sets CORA (500 references), CITESEERX (200 references) and FLUX-CIM (300 references, CS domain)[4] build the source of the evaluation data set. All three data sets consist of a listing of single references without the context of the original reference section. Therefore, these data sets are not directly applicable for the presented approach. A simple script was developed in order to reconstruct reference sections as they would occur in real publications using only references originated in the available data sets. Due to the simplicity of the assignment script and the distribution of the reference styles in the dataset a considerable amount of references could not be assigned to a paper. The resulting data set D_{Paper} contains 28 documents and overall 452 references and resembles reference sections of real papers. Therefore, our data set can be considered more natural. Some erroneous labels and defects due to obvious differences in the annotation guide lines of the three original data sets were corrected. D_{Paper} is randomly splitted into three folds for the evaluation. D^{Train}_{Paper} (315 references, two folds) is used for the training and D^{Test}_{Paper} (137 references) for testing. Additionally, D_{Rest} contains 350 randomly selected references of the remaining references of the original data sets.

3.2 Features

Similar to previous studies with CRFs we use features indicating the capitalization, the length of tokens, numbers, whitespaces on the left and on the right of the observed token, the relative position inside the reference string, n-gram prefixes, n-gram suffixes, as well as the covered text of the token and the covered text of tokens on the left and on the right of the observed token. These features are integrated as normal feature functions. Additionally, annotation-based feature functions, previously denoted by $\mathbb{T}_{\text{feat}}$, for token classes and combinations of tokens are applied. To these belong different usages of punctuations, regular expressions for URLs and simple combinations of features, for example a first name and a capitalized word. Dictionaries for first names, stop words, locations, keywords, journals and publishers were created and added to the annotation-based feature functions. Overall, the applied features are comparable to previously published approaches.

3.3 Settings

Overall, three CRFs are trained for the experimental study: BOOTSTRAP, META and COMPARE. All of them were relying on the same features described in section 3.2. The overall process is build upon UIMA and the ClearTK framework. The machine learning toolkit Mallet is used for an implementation of the CRFs. The presented process model contains two CRFs. The CRF of the bootstrap phase (BOOTSTRAP) represents a simple model for the extraction task. It is trained on the data set D_{Rest} and 250 iteration

[4] all three data sets are available online, e.g., at `http://wing.comp.nus.edu.sg/parsCit/`

	D	.	R	.	Engler	,	M	.	F	.	Kaashoek	,	J	.	W	.	O	'	Toole	.	Exokernel	:	An	Operating	System	Architecture	for	Application	Level	...
Bootstrap result	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	T	T	T	T	T	T	T	T
$\text{END}^*_{\text{AUTHOR}}$																						x								
PM	x		x		x		x		x		x		x		x				x			x								
PERIOD	x		x				x		x				x		x															
COLON																						x								
PeriodSep																			x											
$\text{END}_{\text{AUTHOR}}$																			x											
$\text{END}^t_{\text{AUTHOR}}$																			x											

	S	.	Seneff	and	J	.	Polifroni	.	A	new	restaurant	guide	conversational	system	:	Issues	in	rapid	prototyping	for	specialized	domains	.	In	Pr...					
Bootstrap result	A	A	A	A	A	A	A	A	A	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	B	B		
$\text{END}^*_{\text{AUTHOR}}$									x																					
PM	x				x				x								x								x					
PERIOD	x				x				x																x					
COLON																	x													
PeriodSep									x																x					
$\text{END}_{\text{AUTHOR}}$									x																x					
$\text{END}^t_{\text{AUTHOR}}$									x																					

Figure 2: Two exemplary references with the initially extracted information, the some given features and the assigned meta features. The occurrence of a feature is indicated with "x" and the label of a token is denoted with first letter of the label.

were applied. The CRF of the meta phase (META) has access to the additional meta features $\mathbb{T}_{\text{meta-feat}}$. The model is trained on the data set D^{Train}_{Paper} with unlimited iterations and the threshold β for the meta features is set to 0.01. An additional CRF (COMPARE) is also trained on the data set D^{Train}_{Paper} with the settings of META in order to compare the increase of accuracy due to the meta features. A gaussian variance of 10 is used and the markov order is set to one for all CRFs. The two CRFs BOOTSTRAP and META are trained on two different data sets. The meta features need to be created on real results and not on the almost perfectly labeled data of a training process. If the meta features are only created for correct results in the meta phase, then the advantages for the contextual reuse and correction of the initially extracted information are forfeited. The training of the meta phase needs to rely on realistic and therefore not perfect results for a suitable integration of the meta features.

3.4 Performance Measure

The performance of the presented approach is measured with commonly used methods of the domain. For a field label $l \in \mathbb{T}_{\text{label}}$, let $\text{tp}(l)$ be the number of true positive classified tokens for the label l and define $\text{fn}(l)$ and $\text{fp}(l)$ respectively for false negatives and false positives. Since punctuations contain no information in this domain, only alpha-numeric tokens are considered.

Precision, *recall*, F_1 and *average* F_1 are computed by

$$\text{precision}(l) = \frac{\text{tp}(l)}{\text{tp}(l) + \text{fp}(l)},$$

$$\text{recall}(l) = \frac{\text{tp}(l)}{\text{tp}(l) + \text{fn}(l)},$$

$$F_1(l) = \frac{2 \cdot \text{precision}(l) \cdot \text{recall}(l)}{\text{precision}(l) + \text{recall}(l)},$$

$$\text{Average} = \frac{1}{|\mathbb{T}_{\text{label}}|} \sum_{l \in \mathbb{T}_{\text{label}}} F_1(l).$$

The *instance accuracy* measures how many references have been perfectly classified

$$\text{Instance} = \frac{\#\text{references without an error}}{\#\text{all references}}.$$

3.5 Results

Table 2 contains the results of the experimental study. The second column lists the true positives $\text{tp}(l)$ of each field $l \in \mathbb{T}_{\text{label}}$ and the remaining columns contain the F_1 scores of the three evaluated CRFs BOOTSTRAP, COMPARE and META tested on the data set D^{Test}_{Paper}. The average F_1 and the instance accuracy are added in the last two rows. As mentioned before, the amount of true positives is much smaller than the number of tokens since only alpha-numeric tokens are considered in the evaluation. The information of a date field, for example, is independent of surrounding parentheses or punctuation marks. There are no values added for the editor and note fields. Both fields consist only of a few tokens and achieved an F_1 score of 100.0 in most of the evaluation runs. However, a F_1 score of 0.0 was also sometimes obtained dependent on the distribution of examples in the two data sets D^{Train}_{Paper} and D^{Test}_{Paper}. Therefore, both fields are not considered in the calculation of the average F_1.

Table 1: Results of the evaluation of the three CRFs. The average F_1 is computed without the editor and note field.

	tp	BOOTSTRAP CRF	COMPARE CRF	META CRF
Author	821	99.0	99.1	**99.5**
Booktitle	670	94.8	95.1	**97.5**
Date	200	95.4	**98.0**	97.8
(Editor)	7	0/100	0/100	0/100
Institution	86	32.7	**97.1**	95.1
Journal	186	96.8	89.0	**98.1**
Location	51	86.4	91.7	**92.6**
(Note)	3	0/100	0/100	0/100
Pages	222	90.7	97.5	**97.7**
Publisher	33	87.5	**98.5**	93.7
Tech	75	37.0	87.4	**94.4**
Title	1064	97.0	96.6	**98.3**
Volume	84	98.8	85.1	**98.8**
Average*		83.3	94.1	**96.7**
Instance		75.9	78.8	**85.4**

The CRF META achieved an instance accuracy of 85.4% and an average F_1 score of 96.7%. Compared to the CRF BOOTSTRAP, the error of the instance accuracy was reduced by 39.4% and the error of the average F_1 by 80.1%. Compared to the CRF COMPARE that was trained on the same data set, the error of the instance accuracy was reduced by 31.1% and the error of the average F_1 by 44.1%.

A closer look at the single fields of META and COMPARE reveals that the meta phase was able to improve eight fields and worsened the results of only three fields. Two of these fields, the location and the publisher, contain less true positives than the other fields and strongly depend on dictionaries. The difference in the date fields is caused by only one single misclassified token. Overall, META created 3435 true positives, whereas COMPARE classified 3376 true positives.

3.6 Discussion

The combination of two CRFs and analysis of the local consistencies achieves better results than a single CRF, the state of the art method in the domain of reference extraction. Although the result of the bootstrap phase is mediocre, the local model and the projection of its knowledge are robust enough to create valuable meta features. Hence, the meta phase is able to outperform the commonly applied process. A closer look at the extraction results reveals that the presented approach still trails behind its own potential. The combination of features and meta features often create a situation where a correct classification is obvious. Many false positives and false negatives should not occur with the available features at hand. The boundaries of the author field, for example, are perfectly defined by the created separator features, but the CRF still labels some tokens of the author erroneously. Therefore, the presented approach still provides enormous potential for improvements. A different information extraction technique might integrate the knowledge about the local consistencies better in the meta phase than CRFs. Furthermore, a direct combination of both CRFs in the learning process or an improved projection of the meta features can improve our process model.

The effect of the presented approach on unknown style guides should be investigated in detail. The test data set D_{Paper}^{Test} already contains references with style guides that aren't present in the training data set D_{Paper}^{Train}. However, a test data set only containing unknown styles can illustrate the advantages of our approach compared to the common process model furthermore. A comparison to the results of related publications is problematic. Although the instances of the applied data set were used in previous evaluations, the results can hardly be compared as three different data sets were mixed and some references are left out.

4 Related Work

The extraction of references is an active field of research. Techniques based on Hidden Markov Models, Maximum Entropy Models and Support Vector Machines and several approaches using CRF were published. Peng and McCallum [Peng and McCallum, 2004] established CRF as the state of the art approach for the reference extraction task. They used 350 references of the CORA data sets for training and 150 references for the evaluation. Councill et al. [Councill *et al.*, 2008] applied CRF in their ParseCit system on the CORA data set and evaluated their approach with a 10 fold cross evaluation. In addition, they evaluated also the data sets CITESEERX and FLUX-CIM. Both approaches achieved an average F_1 score of ≈92% and a modified average F_1 score of ≈93%. Table 2 contains details of their evaluation results.

Ng [Ng, 2004] has built the first version of ParsCit. It was based on the Maximum Entropy paradigm and the accuracy was worse compared to the performance of the current system. However, Ng identified different categories of flaws in his extraction process and applied an additional phase for their correction. One step of these repairs processed repeating fields of a single reference, e.g., the occurrence of multiple titles. For the correction of this error, he created a list of all sequences of fields within the extracted references. Then, the multiple fields were resolved using the sequence that occurred most. With all repairs applied, the instance accuracy was increased from 45.6% to 60.8% on the CORA dataset. Compared to our approach, Ng applied only one specialized repair on the sequences in a post processing step in order to correct an error that can be prevented by applying a CRF instead. Furthermore, the evaluation with the CORA data sets itself prevents any statements about improvements by the usage of local consistencies.

Table 2: Results of related publications. In addition to the average F_1 score, the *Average** is computed without the editor and note fields.

	[Peng and McCallum, 2004]	[Councill *et al.*, 2008]		
	CORA	CORA	CITESEERX	FLUX-CIM
Author	99.4	99	96	99
Booktitle	93.7	93	81	97
Date	98.9	99	94	97
Editor	87.7	86	67	-
Institution	94.0	89	74	-
Journal	91.3	91	83	89
Location	87.2	93	85	89
Note	80.8	65	29	-
Pages	98.6	98	91	97
Publisher	76.1	92	81	85
Tech	86.7	86	73	-
Title	98.3	97	93	96
Volume	97.8	96	87	92
Average	91.5	91.1	79.5	93.4
Average*	92.9	93.9	85.3	93.4
Instance	77.3	-	-	-

There are also some knowledge engineering approaches. Cortez et al. [Cortez *et al.*, 2007] evaluated their unsupervised lexicon-based approach on data sets of the domains health science and computer science. An automatically generated, domain-specific knowledge base is applied after a chunking of each text segment in order to identify the fields. Day et al. [Day *et al.*, 2007] created templates for well-known reference styles and used them to extract the fields in journal articles. They achieved an average accuracy of 92.4% on complete fields.

Kluegl et al. [Kluegl *et al.*, 2010] proposed a local adaptive extraction of references with handcrafted transformation rules. A simple model extracts initial fields. The local patterns of these information are stored in a short term memory and are used to create a description of the style guide of the reference section. Then, transformation rules match on a meta level and provide an automatic adaption on the internal previously unknown consistency of the document. The approach is evaluated only on the fields author, title, editor and date. They achieved an average F_1 score of 97.6% on the complete CORA data set and an average F_1 score of 99.7% on natural reference sections based on the CORA data set. This rule-based work on local adaptivity is the basis of our approach. We continued and extended the previous work by two major points: We exchanged the la-

borious knowledge engineering in all stages and combined an automatic creation of the local model with state of the art techniques. Only the definition of suitable features in the feature extraction requires manual effort of a knowledge engineer. Additionally, all 13 possible fields of a reference are extracted and evaluated instead of the four fields.

Our process model can be compared to stacked generalization [Wolpert, 1992] where a meta model is learnt using the output of initial, basic models. Sigletos et al. [Sigletos *et al.*, 2003] among others adapted this meta-learning approach of classification for the information extraction task. In contrast to this approach, we apply only a single base model and are not trying to compose a model directly on the initially extracted information. Instead, we are creating new features in order to exploit the patterns of the unlabeled data during the extraction task.

Recently, advances have been made in joint inference [Poon and Domingos, 2007] that combine different steps of the information extraction task. The presented work possesses some simple peculiarities of that approach, e.g., features transfer inference information in one direction. There is also work published on extending or parameterizing the features of discriminative models. Stewart et al. [Stewart *et al.*, 2008], for example, are learning flexible features for the extraction of references.

5 Conclusions

We have presented a novel combination of two CRFs applied on the local adaptive extraction of references. The initial results of the first CRF are exploited to gain information about the local consistencies. Then, the second CRF is automatically adapted to the previously unknown style guide. This is achieved by changing the manifestation of its features dependent on the currently processed reference section. The results indicate a considerable improvement towards the commonly applied process model. We achieved an average F_1 score of 96.7% and an instance accuracy of 85.4% on the test data set.

Several directions merit a further exploration of the presented work. A combined inference of both CRFs and a local model beyond a description by single features might exploit the full potential of the approach. Combinations and sequences of features are able to describe the local consistencies even if the features extraction provides only simple features. Moving further in this direction leads to a knowledge-based local model that is created, e.g., using subgroup discovery techniques. Comparing the presented approach to previous work [Kluegl *et al.*, 2010], a transformation-based correction of the initially extracted information appears to be very suitable for this task and is able to integrate the local model more straightforward. The knowledge engineering effort can be avoided by learning several binary classificators. Support Vector Machines, for example, can be trained to define transformations of the given information dependent on the local model. Our approach is not restricted to the extraction of references. Even greater improvements compared to the state of the art methods are possible in other domains like curriculum vitae and medical patient records (cf. [Kluegl *et al.*, 2009]).

References

[Cortez *et al.*, 2007] Eli Cortez, Altigran S. da Silva, Marcos André Gonçalves, Filipe Mesquita, and Edleno S. de Moura. FLUXCiM: Flexible Unsupervised Extraction of Citation Metadata. In *JCDL '07: Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries*, pages 215–224, New York, USA, 2007.

[Councill *et al.*, 2008] Isaac Councill, C Lee Giles, and Min-Yen Kan. ParsCit: an Open-source CRF Reference String Parsing Package. In *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, 2008. ELRA.

[Day *et al.*, 2007] Min-Yuh Day, Richard Tzong-Han Tsai, Cheng-Lung Sung, Chiu-Chen Hsieh, Cheng-Wei Lee, Shih-Hung Wu, Kun-Pin Wu, Chorng-Shyong Ong, and Wen-Lian Hsu. Reference Metadata Extraction using a Hierarchical Knowledge Representation Framework. *Decis. Support Syst.*, 43(1):152–167, 2007.

[Ferrucci and Lally, 2004] David Ferrucci and Adam Lally. UIMA: An architectural approach to unstructured information processing in the corporate research environment. *Natural Language Engineering*, 10(3/4):327–348, 2004.

[Kluegl *et al.*, 2009] Peter Kluegl, Martin Atzmueller, and Frank Puppe. Meta-level Information Extraction. In Bärbel Mertsching, Marcus Hund, and Muhammad Zaheer Aziz, editors, *KI*, volume 5803 of *Lecture Notes in Computer Science*, pages 233–240. Springer, 2009.

[Kluegl *et al.*, 2010] Peter Kluegl, Andreas Hotho, and Frank Puppe. Local Adaptive Extraction of References. In *33rd Annual German Conference on Artificial Intelligence (KI 2010)*. Springer, 2010. accepted.

[Lafferty *et al.*, 2001] J. Lafferty, A. McCallum, and F. Pereira. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. *Proc. 18th International Conf. on Machine Learning*, pages 282–289, 2001.

[Ng, 2004] Yong Kiat Ng. Citation Parsing using Maximum Entropy and Repairs. Undergraduate thesis, National University of Singapore, 2004.

[Ogren *et al.*, 2008] Philip V. Ogren, Philipp G. Wetzler, and Steven Bethard. ClearTK: A UIMA toolkit for statistical natural language processing. In *UIMA for NLP Workshop at LREC*, 2008.

[Peng and McCallum, 2004] Fuchun Peng and Andrew McCallum. Accurate Information Extraction from Research Papers using Conditional Random Fields. In *HLT-NAACL*, pages 329–336, 2004.

[Poon and Domingos, 2007] Hoifung Poon and Pedro Domingos. Joint Inference in Information Extraction. In *AAAI'07: Proc. of the 22nd National Conference on Artificial intelligence*, pages 913–918. AAAI Press, 2007.

[Sigletos *et al.*, 2003] Georgios Sigletos, Georgios Paliouras, Constantine D. Spyropoulos, and Takis Stamatopoulos. Meta-Learning beyond Classification: A Framework for Information Extraction from the Web. In *Proc. of the Workshop on Adaptive Text Extraction and Mining. The 14th ECML and the 7th Euro. Conf. on PPKD*, 2003.

[Stewart *et al.*, 2008] Liam Stewart, Xuming He, and Richard S. Zemel. Learning Flexible Features for Conditional Random Fields. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(8):1415–1426, 2008.

[Wolpert, 1992] David H. Wolpert. Stacked Generalization. *Neural Networks*, 5:241–259, 1992.

Towards Understanding the Changing Web: Mining the Dynamics of Linked-Data Sources and Entities

[work in progress]

Jürgen Umbrich and **Marcel Karnstedt**
Digital Enterprise Research Institute (DERI)
National University of Ireland, Galway, Ireland
firstname.lastname@deri.org

Sebastian Land
Rapid-I GmbH
land@rapid-i.com

Abstract

A huge amount of content found on the Web is dynamic by its nature, particularly with the rise of Web 2.0 and beyond. This is of special interest for the Semantic Web community, not only but particularly regarding resources on the Linked Open Data (LOD) Web. However, the dataset dynamics of the LOD graph are hardly explored so far. Existing approaches from the traditional HTML Web are not sufficient to mine and discover the dynamics with satisfying accuracy and efficiency, as they do not consider the special characteristics of LOD. We present first initial results on this topic and discuss future steps. First results are obtained by mining the groups of URIs with similar change frequencies and by applying time series techniques as well as clustering techniques.

1 Motivation

At the time of writing, we can find several hundred datasets published as Linked Open Data (LOD) on the Web. The LOD cloud contains up to several billion Resource Description Framework (RDF) triples of machine readable information, describing real world entities (resources) and the relations among them. This data forms a huge directed and labelled graph (resources are nodes, relations are edges), which can be accessed, traversed and consumed by humans and intelligent software agents. Currently, the so formed LOD graph (see Figure 1) contains billions of nodes but only a couple of million edges [1]. However, experts in the field commonly agree that this gigantic graph will even grow faster in the future and become more dense by adding more labelled links between the nodes. New data and links between the data are added either by humans or by machines (e.g., from data converters like Any23[2] or by content management systems like Drupal 7). The LOD research community focuses on the different issues in identifying, linking and publishing this data.

A still nearly unexplored field are the dynamics of the contained data sets. With the term *dataset dynamics* we refer to *content and interlinking changes in the Linked Data graph*. Besides content changes, the dynamics of nodes (i.e., entities) and links are of particular interest. These can be roughly categorised as follows:

[1] http://esw.w3.org/TaskForces/CommunityProjects/LinkingOpenData/DataSets/[Statistics|LinkStatistics]

[2] http://any23.org/

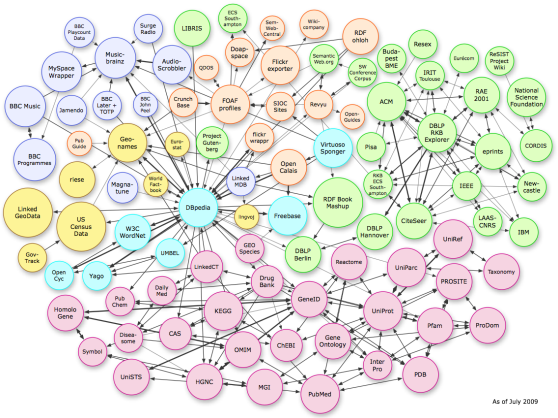


Figure 1: The LOD cloud in mid 2009, courtesy of Cyganiak and Jentzsch.

- **Dynamics of resources**: New nodes are added and old nodes are removed;
- **Dynamics between resources**: New relations are added (in the form of new triples) and old ones might be removed;

In a first attempt [Umbrich *et al.*, 2010] we monitored the change frequency of Linked Data resources and revealed that most of them show the same behavior as HTML Web documents. However, to the best of our knowledge, there is no published work describing attempts to apply techniques from other mature research areas dealing with the analysis and investigation of dataset dynamics. In this work, we take a closer look at how methods from the data mining and machine learning community can be used for this task. We identify suitable and promising techniques, discuss the benefit we expect from their application and present some preliminary results indicating the usefulness.

The motivating use-case for our study of dataset dynamics is to improve concurrent work on an efficient system for performing live queries over the LOD Web [Harth *et al.*, 2010]. The challenge is to decide which (sub-)queries can be run against a cache or data summary and which (sub-)queries have to be executed live over the Web content to guarantee up-to-date results. However, there are several other tasks related to managing Web data that can benefit from a deeper understanding of dataset dynamics: Web crawling and caching [Cho and Garcia-Molina, 2003], maintaining link integrity [Haslhofer and Popitsch, 2009], serving of continuous queries [Pandey *et al.*, 2003].

Regarding the above sketched use cases, there are some main questions we have to focus on. These are:

- What classes of dynamics can and should we distinguish?

- What actual methods are suited to classify LOD resources with respect to their dynamics?
- On what features should these methods be based on?
- How can we efficiently predict future changes and the types of changes?

To start the investigation of these questions we firstly focus on one of the underlying main questions, which we see as the most interesting starting point:

What correlations between resources and their dynamics can we identify using methods from data mining, machine learning and graph analysis?

The specific correlations we expect and therefore plan to investigate are correlations between the dynamics of the resources and:

1. their domain names (i.e., their origin)
2. the used vocabulary (i.e., RDF predicates and classes)
3. their linkage (i.e., if one resource changes how likely is it that resources linked to it change as well)

To achieve this, we first started to continuously monitor a large set of LOD resources. Based on the observed dynamics, we can cluster these resources and apply correlation analysis between sets of resources as well as between resources and their features. In this work, we present initial results on this analysis and discuss future steps.

2 Dataset Dynamics on the LOD Web

In this section, we first introduce the data model of RDF and how it contributes to the LOD Web. Secondly, we discuss how our problem of studying dataset dynamics can be mapped to the problem of the dynamics of nodes in a labelled directed graph. Finally, we elaborate on the importance of data mining to reveal new insights into the dynamics of such a graph.

2.1 RDF and Linked Data

The Resource Description Framework [Manola and Miller, 2004] defines a data format for publishing schema-less data on the Web in the form of $(subject, predicate, object)$ triples. These triples are composed of unique identifiers (URI references), literals (e.g., strings or other data values), and local identifiers called blank nodes as follows:

Definition 1. *(RDF Triple, RDF Term, RDF Graph) Given a set of URI references $\mathcal{U}$, a set of blank nodes $\mathcal{B}$, and a set of literals $\mathcal{L}$, a triple (s, p, o) $\in (\mathcal{U} \cup \mathcal{B}) \times \mathcal{U} \times (\mathcal{U} \cup \mathcal{B} \cup \mathcal{L})$ is called an* RDF triple*, We call elements of $\mathcal{U} \cup \mathcal{B} \cup \mathcal{L}$* RDF terms*. Sets of RDF triples are called* RDF graphs*.*

The notion of graph stems from the fact that RDF triples may be viewed as labelled edges connecting subjects and objects. RDF published on the Web according to the following principles is called *Linked Data* [Berners-Lee, 2006]: 1) URIs are used as names for things: in contrast to the HTML Web where URIs are used to denote content (documents, images), on the Semantic Web URIs can denote entities such as people or cities; 2) URIs should be dereferenceable using the Hypertext Transfer Protocol (HTTP): a user agent should be able to perform HTTP GET operations on the URI; 3) Useful content in RDF should be provided at these URIs: a Web server should return data encoded in one of the various RDF serialisations; 4) Include links to other URIs for discovery: a user agent should be able to navigate from an entity to associated entities by following links, which enables decentralised discovery of new data.

$s_{i,t}$	o_1	o_2	o_3	$\cdots$	o_n
$< URI_1 >$	$s_{1,1}$	$s_{1,2}$	$s_{1,3}$	$\cdots$	$s_{1,n}$
$< URI_2 >$	$s_{2,1}$	$s_{2,2}$	$s_{2,3}$	$\cdots$	$s_{2,n}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
$< URI_m >$	$s_{m,1}$	$s_{m,2}$	$s_{m,3}$	$\cdots$	$s_{m,n}$

Table 1: Change matrix with $s_{i,t} \in \{-1, 0, 1, 2\}$.

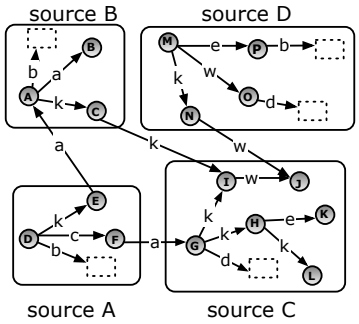


Figure 2: Abstraction of a subgraph of the LOD Web.

As such, Linked Data forms a labelled directed graph as depicted abstractly in Figure 2. We can see that subgraphs are contained in sources (a source is a container of a sub-graph and also has a unique URI, in analogy to Web documents), where resources and sources are interlinked. For the remainder of this paper, we focus on how to analyse the dynamics of the nodes and edges in such a graph. For further simplification, we do not distinguish between changing sources and entities – interested readers are referred to [Umbrich *et al.*, 2010]. We use the following definition of a change of a node:

Definition 2. *(Node state change $s_{i,t}$) A change of the state of a node identified by URI U_i is detected iff the tree of depth 1 with root node U_i differs between two observations o_{t-1}, o_t. We denote a change with $s_{i,t} = 2$, $s_{i,t} = 0$ otherwise. In addition, if a node appears in t we denote this with $s_{i,t} = 1$, if it disappears we use $s_{i,t} = -1$.*

The results of monitoring these changes over time are represented in an $m \times n$ change matrix (m resources and n observations) as illustrated in Table 1.

2.2 Data Mining

Traditionally, change frequencies of Web documents are modeled as Poisson processes [Cho and Garcia-Molina, 2000] and advanced estimators are used to predict the likelihood of the next change [Cho and Garcia-Molina, 2003]. Our current findings uncover that this model holds only for some resources of the LOD Web [Umbrich *et al.*, 2010]. In addition, we strongly believe that by applying more advanced machine learning methods we can perform better predictions of changes. By applying these techniques, we expect to reveal the existance of correlations between the change characteristics of different resources. As such, we will investigate correlation analysis, frequency analysis and techniques for change detection. Concrete techniques that we plan to assess are, among others, SVD and SVM, DFT and wavelet transformation as well as change point detection as known from data streams and graphs.

3 Describing Change Features and Clustering

A first step towards our general objectives is to identify nodes with similar dynamics. Thus, we first investigated how to cluster nodes wrt. to their characteristic dynamics. The main questions we have to answer for that is: What are features describing the dynamics of a resource? Possible describing features are:

- Average change frequency ratio: The dynamics of each node can be simply described by the number of node state changes divided by the overall number of snapshots in the monitored period.
- Statistical summaries of change behaviour: We could use statistical summaries of node state changes, such as central tendencies (arithmetic mean, median or interquartile mean) or statistical dispersion (standard deviation, variance or quantiles).
- Periodicities of changes: such periodicities can be determined by DFT or wavelets transformation, which would overcome obvious issues in using an average change frequency (e.g., one entity changing very often in the beginning of the monitored time but then being rather static, compared to another entity changing regularly in larger intervals over all the time).
- Eigenvalues or principal components: Using reduction techniques like SVD or PCA we can try to extract significant features capturing the characteristics of the dynamics.

Once we decided how to represent dataset dynamics on the basis of the dynamics of nodes we can apply clustering algorithms to group nodes with similar dynamics. Previous experiments indicated the existance of significantly different clusters [Umbrich *et al.*, 2010]. Afterwards, we will use the nodes from the gained clusters to analyse correlations among them and among single nodes and their edges. Methods we have in mind for that are classification approaches and correlation analysis, such as computing the correlation and covariance matrices between URI attributes; e.g., the correlation between node state changes and/or the type of incoming links. We will look into different methods to compute the correlation coefficients; e.g., the Pearson or Pearman's correlation coefficient or entropy-based mutual information/total correlation methods that allow us to detect even more general dependencies. These represent first steps to answer the aforementioned questions about concrete correlations.

4 Preliminary experiments

One question we already investigate in this work (see Section 4.3) is: How many clusters can we identify? Or: What is the "optimal" number of clusters? This is particularly relevant as we first decided to apply k-Means clustering, which requires an a-priori definition of k. If the number of clusters is too small, the clusters contain too many items which are not very similar. If it is too large, we do not capture the actual similarities. With this we want to overcome the limitations of using "soft" categories, e.g., using pre-defined classes like *static*, *low dynamic*, *medium dynamic*, *very dynamic*. This might be sufficient for some applications, but not for the particular approaches we are working on. For instance, we want to be able to decide accurately *when* a crawler has to revisit a site to update an index or how to set a sort of cache coherence time in order to achieve satisfying data freshness. The materials and methods used and applied to get these first preliminary results are described next.

4.1 Data

As the data for our experiments we use a 1% random sample from a data set containing 11 weekly observations of 161K LOD sources crawled using the LDSpider framework [3]. The average observations size is 440MB gzipped and a total number of 2.7M nodes and roughly 7 million links over the whole monitored time. We used the random sample to achieve manageable processing times.

For analysing the sample, we decided to use the data-mining framework RapidMiner[4]. RapidMiner offers a wide variety of data-mining tools, ranging from basic data cleaning to complex transformations and analyses.

4.2 Methods

For detecting node state changes between two observations o_{t-1}, o_t we used a straightforward approach based on a merge-sort scan. After sorting all relevant statements by their syntactic natural order (subject-predicate-object), we performed a pairwise comparison of the statements by scanning two observations in linear time. We record a state change as soon as the order of the statements differed between two observations (e.g., a data producer adds or removes outgoing edges and nodes).

The used sample consists of 26961 URIs and 10 observations (M=26961 $\times$ 10 matrix). In a pre-processing, step we transformed the rectangularly shaped input data into smooth series of values to finally compute the frequency spectrum with a Fourier analysis. The concrete steps involved are:

1. **Organise by change events**: We split the input change matrix into four matrices in the first transformation process. One matrix $M^{ev} = [m^{ev}_{i,t}]; i = 1, 2, \ldots, m; t = 1, 2, \ldots, n$ for each change event ev [ev = -1 (disappear), 0 (no change),1 (appear), 2 (change)]. The matrix values are encoded as follows:
$$m^{ev}_{i,j} = \begin{cases} 1 & \text{if } s_{i,t} = ev \\ 0 & \text{otherwise} \end{cases} \quad (1)$$
2. **Interpolation & Smoothing** We interpolated the data values for each matrix by tenfolding each single value and applying an exponential smoothing over the new time series. The interpolation is necessary since we have only 10 observations for each URI. Further, with the interpolation and smoothing we implicitly model the uncertainty of the event. We only know that a change event occurred between two observations, but not exactly when. This results in four matrices with 100 attributes for each URI. (M^{ev}_{intpol} = 26961 $\times$ 100 matrix)
3. **Fourier Analysis**: After the interpolation and smoothing of the data we performed a Fourier analysis. The analysis resulted in 32 spectrums for each URI ($M^{ev}_{fourier}$ = 26961 $\times$ 32 matrix).
4. **Join of Matrices by their URI**: Finally, we joined the four matrices by their URIs which resulted in our final matrix (M_{final} = 26961 $\times$ 128 matrix).

[3] `http://code.google.com/p/ldspider/`

[4] http://rapid-i.com/content/view/181/190/

4.3 Preliminary Results

Figure 3 shows results analysing the number of clusters for the k-Means clustering with a centroid distance evaluation measure. The high number of different clusters came to our surprise. Clearly, a "soft" category approach with only a handful of clusters as sketched above cannot work out to capture these similarities satisfyingly. Moreover, after a brief manual inspection of the gained clusters, we can intuitively reason about basic correlations, such as nodes from same domains are often found in same clusters. This underlines the appropriateness of the applied methodology and motivates for further work in this direction towards actual correlation analysis. The gained clustering results provide a useful basis for these upcoming tasks.

We also tried to perform an agglomerative clustering. Unfortunately, we could not gain any results with that due to performance issues. Regarding the actually small sample we used, this highlights the need for particularly scalable methods in the context of the huge LOD graph.

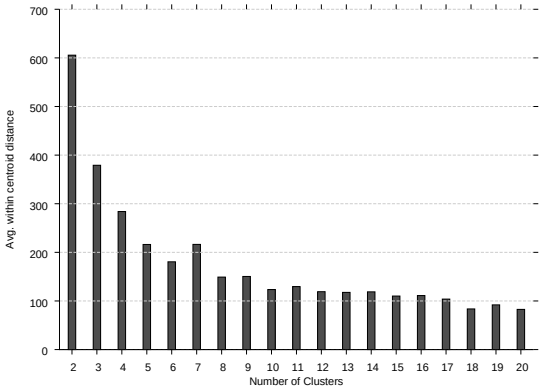


Figure 3: Number of clusters and average centroid distance.

5 Conclusion & Future Work

In this work, we presented our motivation and first ideas to study the dataset dynamics of the LOD Web using machine learning and data-mining approaches. We strongly believe that we can encompass comprehensive details about attributes and features that trigger or relate changes of LOD resources. The gained knowledge can eventually be integrated into a wide range of Web-related applications. In an first attempt, we identified how we can model the time series of changes and analyse its spectrum. Preliminary clustering results indicate the appropriateness of this approach and provide interesting first insights. We hope that this initial report triggers a rich discussion about suited methods and promising approaches in the community.

There exist many directions for future work. Clearly, we need a longer history of changes and thus we will continue our monitoring approach over the next years. Further, we will enrich our change matrix with more information. Possible information includes the incoming labelled links for the nodes, the node state changes of the one hop surrounding nodes and more features about the change event (type and fraction of change). Our future investigations include the following directions:

Change correlations We will investigate siophisticated methods to identify correlations between the dynamics of nodes. Especially, data reduction techniques such as PCA or SVD are of high interest to us and we will explore how and to which extend we can apply them.

Change classification Another area of high relevance for future work is to classify URIs into classes of dynamics. We are particularly interested to find the best features for the classification and we consider to use the outcome of the correlation analysis to increase the classification quality.

Change prediction Knowing in advance at which time in the future a resource is very likely to change is of tremendous value for our use case. Thus, we will explore machine-learning methods to compute the most likely change time based on observed change events. The quality of our predictions can also serve as an evaluation measure of the described approach.

Dealing with incomplete history Eventually, we will explore methods to deal with incomplete change histories. It is hard to monitor a resource constantly over a long time period. In a real world setup, a system will never be able to get a continuous history of change events of a resource. For instance, the window intervals might be too large and multiple changes can happen between to snapshots. In such a case it is of high importance to be able to handle these missing events and still predict and classify accurately.

Acknowledgements

This work based upon works jointly supported by the Science Foundation Ireland under Grant No. SFI/08/CE/I1380 (Lion-2) and under Grant No. 08/SRC/I1407 (Clique: Graph & Network Analysis Cluster) and by the European Community 7th framework ICT-2007.4.4 (No 231519) "e-Lico: An e-Laboratory for Interdisciplinary Collaborative Research in Data Mining and Data-Intensive Science".

References

[Berners-Lee, 2006] Tim Berners-Lee. Linked data, July 2006. `http://www.w3.org/DesignIssues/LinkedData`.

[Cho and Garcia-Molina, 2000] Junghoo Cho and Hector Garcia-Molina. The evolution of the web and implications for an incremental crawler. In *VLDB*, pages 200–209, 2000.

[Cho and Garcia-Molina, 2003] Junghoo Cho and Hector Garcia-Molina. Estimating frequency of change. *ACM Trans. Internet Techn.*, 3(3):256–290, 2003.

[Harth *et al.*, 2010] Andreas Harth, Katja Hose, Marcel Karnstedt, Axel Polleres, Kai-Uwe Sattler, and Jürgen Umbrich. Data summaries for on-demand queries over linked data. In *World Wide Web Conference (WWW '10)*, pages 411–420, 2010.

[Haslhofer and Popitsch, 2009] B. Haslhofer and N. Popitsch. DSNnotify - detecting and fixing broken links in linked data sets. In *DEXA '09 Workshop on Web Semantics (WebS '09)*, 2009.

[Manola and Miller, 2004] Frank Manola and Eric Miller. RDF Primer. W3C Recommendation, February 2004. http://www.w3.org/TR/rdf-primer/.

[Pandey *et al.*, 2003] Sandeep Pandey, Krithi Ramamritham, and Soumen Chakrabarti. Monitoring the dynamic web to respond to continuous queries. In *World Wide Web Conference (WWW '03)*, pages 659–668, 2003.

[Umbrich *et al.*, 2010] Jürgen Umbrich, Michael Hausenblas, Aidan Hogan, Axel Polleres, and Stefan Decker. Towards dataset dynamics: Change frequency of linked open data sources. In *WWW '10 Workshop on Linked Data on the Web (LDOW '10)*, 2010.

Counting-based Output Prediction for Orphan Screening

Katrin Ullrich[1,2], **Christoph Stahr**[1,3], **Thomas Gärtner**[1,4]
[1]Fraunhofer IAIS, Schloss Birlinghoven, 53757 Sankt Augustin
[2]B-IT Research School, [3]University of Magdeburg, [4]University of Bonn
katrin.ullrich@iais.fraunhofer.de, christoph.stahr@student.uni-magdeburg.de,
thomas.gaertner@iais.fraunhofer.de

Abstract

We investigate orphan screening, the search for small molecule ligands of proteins for which no binding ligands are known in advance. Predicting interactions between biologically active molecules is an important step towards effective drug discovery. We propose novel classification and ranking algorithms for orphan screening which are based on counting feature combinations in molecular fingerprints. For the training process we only use positive examples and additional knowledge about the considered proteins and ligands. This knowledge is available in form of protein similarity values a database of molecule compounds. Our algorithms have runtime linear in the number of unlabelled examples.

1 Introduction

We aim at the prediction of ligands for orphan targets. Ligands denote small molecules binding proteins, whereas an orphan target is a protein or protein binding site for which no example ligands are available for learning. More generally, virtual screening denotes the in silico testing of huge databases of molecules for predefined properties, such as the activity against a target. Virtual screening is an important tool to support laboratory testing for the search of ligands. In particular, it is used to preselect promising molecules from the typically very large databases of synthesizable molecules in order to reduce the time and money needed to develop a novel drug. In recent years, machine learning techniques have been very successfully adopted for virtual screening. Whether ligands for the protein under consideration are available beforehand or not, distinguishes the cases of practical relevance. For traditional virtual screening some ligands of the protein are given and others are sought. For orphan screening no ligands of the protein are known and the task is to find some. In the latter case, the protein is called an orphan target.

Virtual screening can be modelled as a classification problem. In contrast to approaches where one is interested in the intensity of the protein ligand bond, in the classification setting we only want to find out whether a small molecule binds a protein or not. Approaches like the one described by Geppert et al. (2008) use support vector machines for non-orphan screening. They employ known ligands as positive training instances and a sample of database molecules as negative training instances.

Building on these techniques, Jacob and Vert (2008) try to improve predictive accuracy with a hypothesis that targets several proteins at the same time. Additionally, they carry their virtual screening method over to orphan screening. In particular, they classify protein-ligand pairs as follows: a pair is considered as positive training instance if the ligand binds to the protein and as negative one, otherwise. Later, a somewhat different technique for both orphan and non-orphan screening was developed by Geppert et al. (2009). There, the authors realize their prediction via a combination of multiple hypotheses. While Geppert et al. (2009) show that their approach is more efficient, both approaches for orphan screening still suffer from two drawbacks: On the one hand, the database molecules, which in fact are molecules for which the label is not known, are assumed to be negative training instances. On the other hand, the amount of available database molecules is huge but the time complexity of these algorithms is cubic in the number of used database molecules.

In this paper we develop a novel approach to virtual screening that is (i) applicable to orphan screening, (ii) does not assume database molecules to be negative instances but models them as unlabelled instances, and (iii) has time complexity linear in the number of database molecules. Our approach is derived from the counting-based structured output prediction algorithm proposed in Gärtner and Vembu (2009). Their method works efficiently if sums over features and pairs of features can be computed in linear time. The term structured output refers to the fact that the results of structured output algorithms are more complex than just labels of natural or real numbers. In our case the structured output will be a fingerprint, a feature vector which represents a compound in a database of molecules. Our aim is to adapt the method of Gärtner and Vembu (2009) for the problem of ligand search, as counting features in fingerprints is computationally easy and may lead to good results in this setting. Actually, we achieve the linear runtime complexity by precomputing the sums of feature vectors and their inner products over the database in linear time.

2 Preliminaries

Small molecules can be represented as bit vectors of zeros and ones in $\mathbb{R}^d$ via molecular fingerprints. The developers of these fingerprints assembled a set of d features in order to display the molecules by means of their chemical properties. Fingerprints have already been established as

standard tools in ligand prediction tasks.

Kernel methods are a popular class of learning algorithms. They include techniques like the support vector machine that are well-founded in learning theory and have been applied in many real-world problems. They work by applying a symmetric and positive semidefinite function $k_{\mathcal{X}} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ as a similarity measure for elements of a space $\mathcal{X}$. A function with these properties is also called kernel function and is known to uniquely generate a reproducing kernel Hilbert space $\mathcal{H}_{\mathcal{X}}$ of real-valued functions defined on $\mathcal{X}$ (see Schölkopf et al. (2001)). In recent years, kernel methods have been extended to handle more complex – so-called structured – outputs. For this purpose we want to learn a joint scoring function f which is defined on the cross-product of two spaces $\mathcal{X}$ and $\mathcal{Y}$. The function f should assign a real score to pairs $(x, y) \in \mathcal{X} \times \mathcal{Y}$ according to "how well x and y fit". Applied to our situation, we want to find ligands for proteins from a set $\mathcal{X}$ by screening a huge set of small molecules stored in a database $\mathcal{Y}$. A common choice, which we will also adopt in this work, is to choose the tensor product of the input and output reproducing kernel Hilbert spaces $\mathcal{H} = \mathcal{H}_{\mathcal{X}} \otimes \mathcal{H}_{\mathcal{Y}}$ as hypothesis space for the joint scoring function. The function space $\mathcal{H}$ has the kernel $k = k_{\mathcal{X}} \otimes k_{\mathcal{Y}}$ given by

$$k\big((x, y), (x', y')\big) = k_{\mathcal{X}}(x, x') \cdot k_{\mathcal{Y}}(y, y'),$$

for $(x, y), (x', y') \in \mathcal{X} \times \mathcal{Y}$, where $k_{\mathcal{X}}$ and $k_{\mathcal{Y}}$ are the kernels of $\mathcal{H}_{\mathcal{X}}$ and $\mathcal{H}_{\mathcal{Y}}$, respectively. We will assume that the feature mapping $\phi : \mathcal{Y} \to \mathbb{R}^d$ is low dimensional and given explicitly. This is indeed naturally the case for virtual and orphan screening where we use fingerprint vectors of length d as molecule representations. In practice, the positive semidefinite kernel $k_{\mathcal{X}}$ is calculated essentially as a DNA sequence identity for two proteins and ranges from 0 to 100. Furthermore, we consider the inner product of feature vectors as the kernel $k_{\mathcal{Y}}$ in the sense of

$$k_{\mathcal{Y}}(y, y') = \langle \phi(y), \phi(y') \rangle,$$

for $y, y' \in \mathcal{Y}$.

3 The Method

In the sequel we propose new loss-functions adapted to the problem of ligand prediction with a database of unlabelled examples and few positive training instances. Concerning the solution of the respective optimization problem we will benefit from the properties of reproducing kernel Hilbert spaces. In fact, our minimizer will turn out to be a linear combination of kernel functions evaluated at the training examples. In addition to orphan screening, our algorithm can also be applied to virtual screening for non-orphan targets, i.e., the search for new ligands of proteins with a few known ligands. The main algorithmic difference between both tasks is the utilization of positive training examples. For orphan screening we exploit ligand information of other proteins and a similarity measure between those proteins (used for training) and the orphan target. Contrary to orphan screening, for virtual screening of non-orphan targets we may additionally use the information we already have for the target in consideration.

3.1 Counting-Based Structured Output Prediction

We are considering a set of proteins $\mathcal{X}$ and a database of small molecules $\mathcal{Y}$, where the latter contains the potential ligands. In general, we are looking for a joint scoring function $f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, which assigns a real value to every pair $(x, y) \in \mathcal{X} \times \mathcal{Y}$. This scoring value should be the higher the better ligand y binds to protein x. The hypothesis space for f shall be the reproducing kernel Hilbert space $\mathcal{H} = \mathcal{H}_{\mathcal{X}} \otimes \mathcal{H}_{\mathcal{Y}}$ with product kernel $k = k_{\mathcal{X}} \otimes k_{\mathcal{Y}}$. Suppose we have positive training examples

$$(x_1, \mathcal{Y}_1), \ldots, (x_m, \mathcal{Y}_m) \in \mathcal{X} \times 2^{\mathcal{Y}},$$

i.e., a set $\mathcal{Y}_i$ of known ligands for every protein $x_i \in \{x_1, \ldots, x_m\} = \mathcal{X}_{train} \subseteq \mathcal{X}$. Moreover, we assume the availability of further knowledge about the candidate space $\mathcal{Y}$. For our algorithm this knowledge consists of a database containing the elements of $\mathcal{Y}$ in form of feature vectors. Note, that we do not consider the exponentially large set of all possible molecule representations. With unlabelled examples we denote every possible combination (x_i, z) such that $x_i \in \mathcal{X}_{train}$ and $z \in \mathcal{Z}_i$, with $\mathcal{Z}_i := \mathcal{Y} \setminus \mathcal{Y}_i$. These pairs are unlabelled examples as we do not know beforehand whether z binds to x_i or not. For orphan screening the considered orphan target $\tilde{x}$ is an element of $\mathcal{X} \setminus \mathcal{X}_{train}$, while a non-orphan target is contained in $\mathcal{X}_{train}$ by construction of the two different tasks. Finally, our goal is a sorted list of the compounds in $\mathcal{Y}$ with respect to a certain protein $\tilde{x}$. Therefore, we want to assign high values of the scoring function f to positive examples and small values to the probably negative unlabelled examples, respectively. We will present different loss-functions which fulfill this requirement (and depend on both positive and unlabelled examples) in the next section. In principle, we want to learn a function $f^* \in \mathcal{H}$ which minimizes the regularized risk functional

$$f^* = \operatorname*{argmin}_{f \in \mathcal{H}} \lambda \|f\|_{\mathcal{H}}^2 + \sum_{i=1}^{m} \mathcal{R}(f, i) \tag{1}$$

where $\mathcal{R}(\cdot, i)$ is the loss-function for the i-th positive training example. Since the quadratic function is strictly monotonically increasing and the regularized risk term $\sum_{i=1}^{m} \mathcal{R}(f, i)$ specified below is a function with inputs (x_i, y) and (x_i, z) such that $x_i \in \mathcal{X}_{train}$, $y \in \mathcal{Y}_i$, and $z \in \mathcal{Z}_i$, $i = 1, \ldots, m$, we may apply the representer theorem (Schölkopf et al., 2001). We obtain for our optimizer f^* that

$$\begin{aligned} f^* \in \mathcal{F} &= \operatorname{span}\Big\{ k_{\mathcal{X}}(\cdot, x) \cdot k_{\mathcal{Y}}(\cdot, z) : x \in \mathcal{X}, z \in \mathcal{Y} \Big\} \\ &= \operatorname{span}\Big\{ k_{\mathcal{X}}(\cdot, x) \cdot \langle \phi(\cdot), \phi(z) \rangle : x \in \mathcal{X}, z \in \mathcal{Y} \Big\}, \end{aligned}$$

where $k_{\mathcal{X}}(\cdot, \cdot)$ is the adequate protein similarity value, and $\langle \phi(\cdot), \phi(\cdot) \rangle$ is the inner product in the output space with dimension d. Usually, $\mathcal{Y}$ is given as a huge database of instances. For this reason we would like to scale down $\mathcal{F}$. In practice the dimension d of the feature space is often much smaller than the number of elements in $\mathcal{Y}$ (e.g. fingerprint *maccs* has $d \approx 160$, while $|\mathcal{Y}|$ is usually greater than 100.000). If we consider the canonical orthonormal

basis $e_1, \ldots, e_d$ of $\mathbb{R}^d$, the transformation

$$\begin{aligned}\mathcal{F} &= \text{span}\left\{k_{\mathcal{X}}(\cdot, x) \cdot \langle \phi(\cdot), \phi(z)\rangle : x \in \mathcal{X}_{train}, z \in \mathcal{Y}\right\} \\ &= \text{span}\left\{k_{\mathcal{X}}(\cdot, x) \cdot \langle \phi(\cdot), \sum\nolimits_{l=1}^{d} \beta_l^z e_l\rangle : \right. \\ &\qquad\qquad \left. x \in \mathcal{X}_{train}, z \in \mathcal{Y}, \beta_l^z \in \mathbb{R}\right\} \\ &= \text{span}\left\{ \sum\nolimits_{l=1}^{d} \beta_l^z k_{\mathcal{X}}(\cdot, x) \cdot \langle \phi(\cdot), e_l\rangle : \right. \\ &\qquad\qquad \left. x \in \mathcal{X}_{train}, z \in \mathcal{Y}, \beta_l^z \in \mathbb{R}\right\}\end{aligned}$$

shows that

$$f^*(x, y) = \sum_{i=1}^{m} \sum_{l=1}^{d} \alpha_{l,i} k_{\mathcal{X}}(x, x_i) \langle \phi(y), e_l\rangle \qquad (2)$$

for appropriate coefficients $\alpha_{l,i} \in \mathbb{R}$. Now we have to optimize f^* over only $m \cdot d$ coefficients (instead of $m \cdot |\mathcal{Y}|$ coefficients as in the initial representation of $\mathcal{F}$). We will find the solution f^* of (1) by minimizing with respect to $\alpha \in \mathbb{R}^{d \times m}$. Following, we propose new loss-functions for ligand prediction which finally determines the optimization problem (1) completely.

3.2 Loss-Functions for Orphan Screening

As mentioned above, we want to learn a function f that maps positive examples to higher values than unlabelled ones. For the associated optimization problem we present different loss-functions and contrast them with each other. The loss-functions are all based on quadratic loss (compare Gärtner and Vembu (2009)). They vary in the treatment of labelled and unlabelled examples, respectively.

Suppose we have a set $\mathcal{X}_{train} = \{x_1, \ldots, x_m\}$ of proteins and a database of molecules $\mathcal{Y}$. Furthermore, for every $x_i \in \mathcal{X}$ we have a set $\mathcal{Y}_i$ of already verified ligands and the complement set of unlabelled database molecules $\mathcal{Z}_i$ with respect to x_i. We will refer to $(x_i, \mathcal{Y}_i)$ as the i-th training unit.

We propose to minimize the regularized risk functional (1) with regularization constant $\lambda > 0$ and training units $1, \ldots, m$, for the following loss-functions $\mathcal{R}_1, \ldots, \mathcal{R}_4$

(ranking simple)
$$\mathcal{R}_1(f, i) = \sum_{y \in \mathcal{Y}_i} \sum_{z \in \mathcal{Z}_i} [1 - f_i(y) + f_i(z)]^2$$

(classification simple)
$$\mathcal{R}_2(f, i) = \sum_{y \in \mathcal{Y}_i} [1 - f_i(y)]^2 + \tfrac{1}{|\mathcal{Z}_i|} \sum_{z \in \mathcal{Z}_i} [1 + f_i(z)]^2$$

(ranking mean)
$$\mathcal{R}_3(f, i) = \sum_{y \in \mathcal{Y}_i} \left[1 - f_i(y) + \tfrac{1}{|\mathcal{Z}_i|} \sum_{z \in \mathcal{Z}_i} f_i(z)\right]^2$$

(classification mean)
$$\mathcal{R}_4(f, i) = \sum_{y \in \mathcal{Y}_i} [1 - f_i(y)]^2 + \left[1 + \tfrac{1}{|\mathcal{Z}_i|} \sum_{z \in \mathcal{Z}_i} f_i(z)\right]^2,$$

where $f_i(y) := f(x_i, y)$.
The loss-functions $\mathcal{R}_1, \ldots, \mathcal{R}_4$ are categorized according to their treatment of unlabelled examples (*simple/mean*) and to their arrangement of examples via classification or ranking (*classification/ranking*). The functions $\mathcal{R}_2$ and $\mathcal{R}_4$ push the positive and unlabelled examples for every protein $x_i \in \mathcal{X}$ to scores of $+1$ or -1, respectively. In contrast, $\mathcal{R}_1$ and $\mathcal{R}_3$ rank ligands in an independent fashion for every training example $x_i \in \mathcal{X}_{train}$. Another qualitative differentiating feature is that $\mathcal{R}_1$ and $\mathcal{R}_2$ are constructed such that every single unlabelled example is ranked or classified. In contrast, $\mathcal{R}_3$ and $\mathcal{R}_4$ handle unlabelled examples as a unit. Only the mean of their values of f is supposed to be lower than the values of f for positive examples. For our data setting we would expect the classification mean approach to work best. We believe this because, on the one hand, we have classification-type data. On the other hand, it should be more appropriate to pull the mean of the values of the scoring function for unlabelled examples down. Actually, the unlabelled examples contain ligands which, by construction, are supposed to have high values of f.

An interesting property of the optimization problems above is that they are equivalent to using the second order Taylor approximation of the exponential loss. For example, consider

$$\min \lambda \|f\|^2 + \sum_{i=1}^{m} \sum_{y \in \mathcal{Y}_i} \sum_{z \in \mathcal{Z}_i} [1 - f_i(y) + f_i(z)]^2$$

and

$$\min \lambda \|f\|^2 + \sum_{i=1}^{m} \sum_{y \in \mathcal{Y}_i} \sum_{z \in \mathcal{Z}_i} \exp[1 - f_i(y) + f_i(z)].$$

Both optimization problems are equivalent except for a removal of multiplicative or additive constants resulting in a shift of the objective function.
In the sequel we will sketch how to obtain compact formulas for all functions necessary for optimization.

3.3 Objective Function and Algorithm

In order to solve the optimization problem in (1), according to the parameterization given by (2), we have to minimize with respect to $\alpha \in \mathbb{R}^{d \times m}$. Let us define the terms $Y \in \mathbb{R}^{m \times d}$, $C \in \mathbb{R}^{d \times d}$, and $\Phi \in \mathbb{R}^d$ by

$$Y_{i\cdot} := \sum_{y \in \mathcal{Y}_i} \phi^T(y)\,,\; C := \sum_{z \in \mathcal{Y}} \phi(z)\phi(z)^T\,,\; \Phi := \sum_{z \in \mathcal{Y}} \phi(z),$$

the kernel matrix of protein similarities, $[K]_{i,j=1}^{m} := k_{\mathcal{X}}(x_i, x_j)$, and the vectors of constants

$$\begin{aligned} V &:= (|\mathcal{Y}_1|, \ldots, |\mathcal{Y}_m|)^T\,, \\ W &:= (|\mathcal{Y}| - 2|\mathcal{Y}_1|, \ldots, |\mathcal{Y}| - 2|\mathcal{Y}_m|)^T\,, \\ S &:= \left(\frac{|\mathcal{Y}_1|}{|\mathcal{Z}_i|^2}, \ldots, \frac{|\mathcal{Y}_m|}{|\mathcal{Z}_m|^2}\right)^T, \\ \overline{S} &:= \left(\frac{1}{|\mathcal{Z}_i|^2}, \ldots, \frac{1}{|\mathcal{Z}_m|^2}\right)^T, \\ U &:= \left(\frac{|\mathcal{Y}_1|}{|\mathcal{Z}_i|}, \ldots, \frac{|\mathcal{Y}_m|}{|\mathcal{Z}_m|}\right)^T, \overline{U} := \left(\frac{1}{|\mathcal{Z}_i|}, \ldots, \frac{1}{|\mathcal{Z}_m|}\right)^T. \end{aligned}$$

As the definitions of C and Φ include counts of fingerprint features or feature combinations in linear time, it becomes clear why our method is called counting-based. These constants can be computed in linear time by a single pass

over the database. With this constants we obtain a compact formulation (independent of the size of the database) of the objective functions $o(\alpha, \mathcal{R}_1), \ldots, o(\alpha, \mathcal{R}_4)$ with the respective loss-functions $\mathcal{R}_1, \ldots, \mathcal{R}_4$

$$
\begin{aligned}
o(\alpha, \mathcal{R}_1) &= \lambda \operatorname{tr}(\alpha K \alpha^T) - |\mathcal{Y}| \operatorname{tr}(Y \alpha K) + V^T K \alpha^T \Phi \\
&\quad + \frac{1}{2} \sum_{i=1}^{m} W_i (\alpha K_{\cdot i})^T C_{\mathcal{Y}_i} (\alpha K_{\cdot i}) \\
&\quad + \frac{1}{2} V^T \operatorname{diag}(K \alpha^T C \alpha K) \\
&\quad - \Phi^T \alpha K \operatorname{diag}(Y \alpha K) + \| \operatorname{diag}(Y \alpha K) \|^2 \\
o(\alpha, \mathcal{R}_2) &= \lambda \operatorname{tr}(\alpha K \alpha^T) - (1 + \overline{U})^T \operatorname{diag}(Y \alpha K) \\
&\quad + \overline{U}^T K \alpha^T \Phi + \frac{1}{2} \overline{U}^T \operatorname{diag}(K \alpha^T C \alpha K) \\
&\quad + \frac{1}{2} \sum_{i=1}^{m} (1 - \overline{U})_i (\alpha K_{\cdot i})^T C_{\mathcal{Y}_i} (\alpha K_{\cdot i}) \\
o(\alpha, \mathcal{R}_3) &= \lambda \operatorname{tr}(\alpha K \alpha^T) - (\mathbf{1} + U)^T \operatorname{diag}(Y \alpha K) \\
&\quad + U^T K \alpha^T \Phi + \frac{1}{2} \sum_{i=1}^{m} (\alpha K_{\cdot i})^T C_{\mathcal{Y}_i} (\alpha K_{\cdot i}) \\
&\quad + \frac{1}{2} \Phi^T \alpha K (S \circ K \alpha^T \Phi) \\
&\quad - \Phi^T \alpha K ((S + \overline{U}) \circ \operatorname{diag}(Y \alpha K)) \\
&\quad + \left\| \left(\sqrt{\frac{1}{2} S + \overline{U}} \right) \circ \operatorname{diag}(Y \alpha K) \right\|^2 \\
o(\alpha, \mathcal{R}_4) &= \lambda \operatorname{tr}(\alpha K \alpha^T) - (\mathbf{1} + \overline{U})^T \operatorname{diag}(Y \alpha K) \\
&\quad + \overline{U}^T K \alpha^T \Phi + \frac{1}{2} \sum_{i=1}^{m} (\alpha K_{\cdot i})^T C_{\mathcal{Y}_i} (\alpha K_{\cdot i}) \\
&\quad + \frac{1}{2} \Phi^T \alpha K (\overline{S} \circ K \alpha^T \Phi) \} \\
&\quad - \Phi^T \alpha K (\overline{S} \circ \operatorname{diag}(Y \alpha K)) \\
&\quad + \left\| \left(\sqrt{\frac{1}{2} \overline{S}} \right) \circ \operatorname{diag}(Y \alpha K) \right\|^2 .
\end{aligned}
$$

With standard techniques we also obtained the gradient of the objective function and the product of the Hessian with a vector $v \in \mathbb{R}^{d \times m}$ which we need for optimization. Due to the positive semidefinite kernel matrix K and the particular structure of the objective function, we deal with a convex optimization problem. Hence, we can use the Newton conjugate gradient method to obtain a solution for our minimization problems.

Our ligand prediction algorithm works as follows for a given target protein x: After learning the scoring function f we calculate the values $f(x, z)$ for every database molecule $z \in \mathcal{Z}_x$ (note, orphan screening and virtual screening for non-orphan targets only vary in the data used for training). Afterwards, all those molecules are sorted with respect to their score. The performance of the algorithm can be measured by recovery rates, i.e., the number of actual ligands among the first s sorted molecules (compare also Geppert et al., 2009), where s is an appropriate threshold.

4 Conclusion and Future Work

In this paper we developed several counting-based structured output prediction algorithms for orphan screening as well as traditional virtual screening. For that we established four loss-functions evaluating the quality of a scoring function f. The algorithms aim at an assignment of scores to small molecules in a database according to how good they bind a target protein, in particular, an orphan target. The loss-functions are constructed such that: After a sorting of the database with respect to the scoring values, ligands of the target protein should be in the top positions of the sorted list of molecules. Furthermore, the design of the loss-functions shows our intention to model the database molecules as unlabelled examples.

We will measure the accuracy of our approach in real-world virtual as well as orphan screening settings. Concretely, we want to test our four methods on a set of proteins, each with a set of known ligands. Furthermore, we plan to vary the algorithm. For example, we want to include actual negative examples, insert an additional variance term for the neutral examples, or invert input and output space. Moreover, an important part of our practical studies will be the comparison of our algorithm with SVMStruct and baseline orphan screening methods. In addition to the classification setting, we want to test regression approaches.

Acknowledgments

This work was supported in part by the German state NRW within the B-IT Research School and by the German Research Foundation (DFG) under the reference number 'GA 1615/1-1'.

References

D. Erhan, P.-J. L'Heureux, S. Y. Yue, and Y. Bengio. Collaborative filtering on a family of biological targets. *Journal of Chemical Informatics and Modelling*, 46: 626–635, 2005.

T. Gärtner and S. Vembu. On structured output training: hard cases and an efficient alternative. *Machine Learning Journal (Special Issue of ECML PKDD)*, 76(2):227–242, 2009.

H. Geppert, T. Horváth, T. Gärtner, S. Wrobel, and J. Bajorath. Support-vector-machine-based ranking significantly the effectiveness of similarity searching using 2d fingerprints and multiple reference compounds. *Journal of Chemical Informatics and Modelling*, 48:742–746, 2008.

H. Geppert, J. Humrich, D. Stumpfe, T. Gärtner, and J. Bajorath. Ligand prediction from protein sequence and small molecule information using support vector machines and fingerprint descriptors. *Journal of Chemical Informatics and Modelling*, 49:767–797, 2009.

L. Jacob and J.-P. Vert. Protein-ligand interaction prediction: an improved chemogenomics approach. *Bioinformatics*, 24(19):2149–2156, 2008.

B. Schölkopf, R. Herbrich, and A. J. Smola. A Generalized Representer Theorem. In *Computational Learning Theory*, volume 2111 of *Lecture Notes in Computer Science*, pages 416–426. Springer, Berlin, 2001.

Workshop Information Retrieval WIR 2010

Ingo Frommholz
University of Glasgow
Scotland, UK

Claus-Peter Klas
Fernuniversität Hagen
Germany

The IR Workshop

The ubiquity of search systems has led to the application of information retrieval technology in many new contexts (e.g. mobile and international) and for new object types (products, patents, music). To develop appropriate products, basic knowledge on information retrieval needs to be revisited and innovative approaches need to be applied, for example by allowing for user interaction or by taking the user's situational context into account. The quality of information retrieval needs to be evaluated for each context. Large evaluation initiatives respond to these challenges and develop new benchmarks.

The workshop Information Retrieval 2010 of the Special Interest Group on Information Retrieval within the German Gesellschaft für Informatik (GI) provides a forum for scientific discussion and the exchange of ideas. The workshop takes place in the context of the LWA "Learning, Knowledge and Adaptivity" workshop week (LWA, Oct 4-6, 2010) at the University of Kassel in Germany. This workshop continues a successful series of conferences and workshops of the Special Interest Group on Information Retrieval. The workshop addresses researchers and practitioners from industry and universities. Especially Doctorate and Master students are encouraged to participate.

WIR 2010

The following types of submissions were requested this year:

- Full Papers (8 accepted submissions)
- Short Papers (2 accepted submissions): Position papers or work in progress
- Poster and Demonstrations (1 accepted submissions): Poster and presentation of ideas, systems or prototypes

The following areas were covered by the workshop:

- Development and optimization of retrieval systems
- Retrieval with structured and multimedia documents
- Evaluation and evaluation research
- Text mining and information extraction
- Multilingual systems
- Digital libraries
- User interfaces and user behavior
- Interactive IR
- Combination of structured and unstructured search
- Machine learning in information retrieval
- Information retrieval and knowledge management
- Information retrieval and the semantic web
- Search Engine Optimization
- Social Search

Program committee

The program committee had the following members (in alphabetical order):

- Prof. Dr. Maximilian Eibl, TU Chemnitz
- Prof. Dr. Reginald Ferber, Hochschule Darmstadt
- Dr. Ingo Frommholz, University of Glasgow, UK (Chair)
- Prof. Dr. Norbert Fuhr, Universität Duisburg-Essen
- René Hackl, FIZ Karlsruhe
- Prof. Dr. Andreas Henrich, Universität Bamberg
- Frank Hopfgartner, University of Glasgow, UK
- Dr. Claus-Peter Klas, Fernuniversität Hagen (Chair)
- Dr. Michael Kluck, Stiftung Wissenschaft und Politik, Berlin
- Dr. Sascha Kriewel, Universität Duisburg-Essen
- Dr. Johannes Leveling, Dublin City University, Ireland
- Dr. Thomas Mandl, Universität Hildesheim
- Dr. Wolfgang Müller, EML Research
- Dr. Günter Neumann, DFKI
- Prof. Dr. Vivien Petras, Humboldt-Universität, Berlin
- Prof. Dr. Marc Rittberger, DIPF, Frankfurt am Main
- Dr. Thomas Roelleke, Queen Mary University of London, UK
- Dr. Ralf Schenkel, Universität des Saarlandes, Saarbrücken
- Prof. Dr. Ingo Schmitt, TU Cottbus
- Prof. Dr. Benno Stein, Universität Weimar
- Dr. Ulrich Thiel, Fraunhofer IPSI, Darmstadt
- Prof. Dr. Gerhard Weikum, Max-Planck-Institut für Informatik
- Prof. Dr. Judith Winter, Fachhochschule Frankfurt am Main
- Prof. Dr. Christian Wolff, Universität Regensburg
- Prof. Dr. Christa Womser-Hacker, Universität Hildesheim

We would like to thank the authors for their submissions, and we also thank the members of the program committee for providing helpful constructive reviews.

Kassel, October 2010,

Ingo Frommholz and Claus-Peter Klas

Language Models, Smoothing, and IDF Weighting

Najeeb Abdulmutalib, Norbert Fuhr
University of Duisburg-Essen, Germany
{najeeb|fuhr}@is.inf.uni-due.de

Abstract

In this paper, we investigate the relationship between smoothing in language models and idf weights. Language models regard the relative within-document-frequency and the relative collection frequency; idf weights are very similar to the latter, but yield higher weights for rare terms. Regarding the correlation between the language model parameters and relevance for two test collections, we find that the idf type of weighting seems to be more appropriate. Based on the observed correlation, we devise empirical smoothing as a new type of term weighting for language models, and retrieval experiments confirm the general applicability of our method. Finally, we show that the most appropriate form of describing the relationship between the language model parameters and relevance seems to be a product form, which confirms a language model proposed before.

1 Introduction

Since several years, language models are the preferred type of IR models [Hiemstra, 1998; Ponte and Croft, 1998; Berger and Lafferty, 1999]. In contrast to other models, they explicitly include a document indexing model that relates the within-document frequency of a term to its indexing weight. On the other hand, there is no explicit notion of probability of relevance. Closely related to this statement, there is the somewhat unclear relation between tf*idf weighting (like e.g. in the classic vector space model or in BM25) and the probabilistic parameters of language models.

In this paper, we present some empiric results that relate language models to tf*idf weights, which leads us to a new smoothing method giving us good retrieval results.

2 Language Models

Language models regard a text in form of a sequence of words as a stochastic process. Thus, for a given vocabulary (set of terms) T, a language model θ is defined as a probability distribution

$$\theta = \{(t_i, P(t_i|\theta)|t_i \in T)\} \quad \text{with} \sum_{t_i \in T} P(t_i|\theta) = 1$$

In the most simple form, one assumes independence of term occurrences, and thus the probability of a document text $d = t_1 t_2 t_3 \ldots t_m$ wrt. to language model θ can be computed as $P(d|\theta) = \prod_{j=1}^{m} P(t_j|\theta)$.

The basic idea for defining a retrieval function is to compare the document's d language models to that of the query q. One way for doing this is to compute the probability that the query was generated by the document's language model:

$$\begin{aligned} P(q|d) &\approx \prod_{t_i \subseteq q^T} P(t_i|d) \\ &= \prod_{t_i \in q^T \cap d^T} P_s(t_i|d) \prod_{t_i \in q^T - d^T} P_u(t_i|d) \\ &= \prod_{t_i \in q^T \cap d^T} \frac{P_s(t_i|d)}{P_u(t_i|d)} \prod_{t_i \in q^T} P_u(t_i|d) \qquad (1) \end{aligned}$$

Here d^T denotes the set of terms occurring in the document, and q^T refers to the set of query terms. $P_s(t_i|d)$ denotes the probability that the document is about t_i, given that t_i occurs (is seen) in the document. On the other hand, $P_u(t_i|d)$ denotes the same probability for those terms t_i not occurring (is unseen) in the document.

The estimation of these parameters suffers from the problem of sparse data. Thus, a number of smoothing methods have been developed. Let F denote the total number of tokens in the collection and $cf(t)$ the collection frequency of term t, $l(d)$ the number of tokens in document d and $tf(t,d)$ the corresponding within-document frequency of term t . Then we estimate

$$P_{avg}(t) = \frac{cf(t)}{F} \qquad \text{and} \qquad P_{ml}(t|d) = \frac{tf(t,d)}{l(d)}$$

where $P_{avg}(t)$ is the average relative frequency of t in the collection, and $P_{ml}(t|d)$ is the maximum likelihood estimate for the probability of observing t at an arbitrary position in d.

Various smoothing methods have been developed in the past. In this paper, we only regard the most popular one, namely Jelinek-Mercer (JM) smoothing [Jelinek and Mercer, 1980]:

$$P_s(t_i|d) = (1-\lambda)P_{ml}(t|d) + \lambda P_{avg}(t) \qquad (2)$$

Here λ (with $0 \leq \lambda \leq 1$) is a global smoothing parameter that allows for collection-specific tuning. For the unseen terms, [Zhai and Lafferty, 2001] propose the following estimate:

$$\begin{aligned} P_u(t_i|d) &= \alpha_d P_{avg}(t) \\ \text{with} \quad \alpha_d &= \frac{1 - \sum_{t_i \in q^T \cap d^T} P_{avg}(t)}{1 - \sum_{t_i \in q^T \cap d^T} P_{ml}(t|d)} \end{aligned}$$

As we can see from eqn 2, the term weight is a weighted sum of its relative within-document frequency P_{ml} and its relative frequency in the whole collection, P_{avg}. Classic tf*idf weighting formulas are based on the same parameters. However, whereas the tf part of these types of weights usually is some monotonic transformation of P_{ml}, the idf part is the negative logarithm of the document frequency, i.e. similar to $-\log(P_{avg})$. (Note, however, that P_{avg} refers to tokens, whereas the idf weight regards the number of documents in which a term occurs. A theoretic treatment of this aspect can be found in [Roelleke and Wang, 2008]. Here we assume that this difference is negligible.) The theoretic justification of idf weights goes back to [Croft and Harper, 1979], who showed that the relative document frequency is an estimate for the probability of the term occurring in a nonrelevant document, thus linking this parameter to relevance. Thus, we have the language model interpretation of P_{avg} on one hand, and the relevance-oriented interpretation of the idf weight on the other hand. In the former, the weight of a term grows monotonically with P_{avg}, whereas the opposite is true for idf weights. Although [Roelleke and Wang, 2008] shows how the two kinds of weightings relate to each other, this paper does not resolve the apparent contradiction.

3 Language model parameters vs. probability of relevance

As an alternative to a theoretic treatment, we performed an empirical study on the distribution of the language model parameters P_{ml} and P_{avg} in relevant and nonrelevant documents. For that, we regarded two test collections:

1. The INEX 2005 collection consisting of 16819 journal articles (764 MB), where we regard each of the 21.6 million XML elements as a document[1]. Due to this document definition, we have a great variation in document lengths, as is illustrated in figure 1. As query set, we use the corresponding 29 content-only queries.
2. The AP part of the TREC collection containing 240,000 documents, along with TREC queries 51-100 and 101-150.

For computing our statistics for the given query sets , we considered all query-document pairs where the document contains at least one query term. In case the document is relevant, all query terms are regarded as being relevant for this document otherwise all terms are irrelevant. Now we aim at relating the (P_{ml}, P_{avg}) pairs of terms to their probability of relevance $P(R|t)$ that a document containing t will be judged relevant to a random query containing t as query term. For that, we perform an appropriate binning of (P_{ml}, P_{avg}) pairs into two-dimensional intervals, and then we compute the ratio of relevant pairs among all pairs in an interval.

[1]Retrieval of XML elements can be used as a first step in a two-stage process for focused XML retrieval, where the second step picks the most specific elements from each XML document that answer the query in the most exhaustive way.

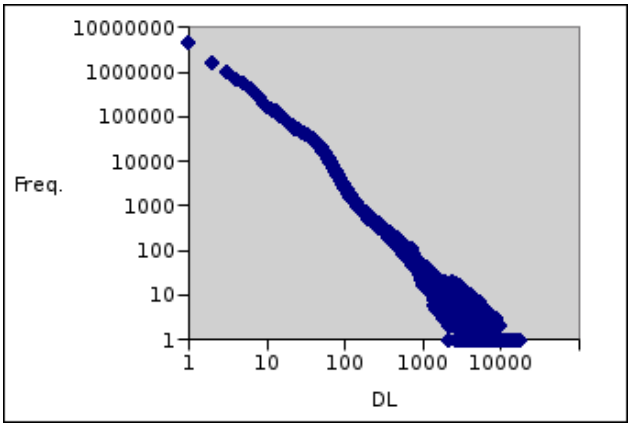


Figure 1: Distribution of document lengths in INEX

Figure 2 shows the corresponding statistics for the INEX collection[2]. At first glance, we already see that this statistics confirms the tf*idf heuristics: the higher P_{ml} and the smaller P_{avg}, the higher $P(R|t)$. Moreover, P_{avg} is dominating and P_{ml} has only a minor effect: For any given P_{avg} interval, the $P(R|t)$ values are roughly all in the same order of magnitude (ignoring the case where $P_{ml} = 0$), whereas for any P_{ml} interval the $P(R|t)$ values vary by several orders of magnitude. This observation contrasts with the standard justification of smoothing methods in language models, where it is said that P_{ml} is the dominating factor and P_{avg} is used only for dealing with data sparsity. The results also show that for $P_{ml} = 0$ (terms not occurring in the document), $P(R|t)$ is much smaller than for $P_{ml} > 0$. For higher values of P_{avg}, $P(R|t)$ seems to be zero. However, using a logarithmic scale, we can see that $P(R|t)$ decreases monotonically when P_{avg} increases.

The corresponding results for the TREC collection are shown in figure 3. The major difference to the TREC collection is that in TREC, the slope in the P_{avg} direction is not as high as in INEX. One possible explanation could be the fact that the relevance definition used in TREC is less strict than the INEX one. Furthermore, for terms not occurring in the document, there is only a minor $P(R|t)$ difference in comparison to those having low P_{ml} values.

Overall, these empirical observations confirm the dominant role of P_{avg} wrt. retrieval quality. This is in stark contrast to the standard language model justification, saying that P_{ml} is more important and P_{avg} only helps in smoothing.

4 Implementing empirical smoothing

Based on the observations described above, we now want to propose a new approach for smoothing, which we call empirical smoothing. The basic idea is already illustrated in figures 2–3: For each possible combination of (P_{ml}, P_{avg}) values of a term, these plots show the corresponding probability $P(R|t)$. So it seems straightforward to use these values as result of the smoothing process.

In principle, there are three different ways for implementing this idea:

Direct use of interval values: As outlined above, we can directly use the probability estimates of $P(R|t)$ from figures 2-3. Thus, given a (P_{ml}, P_{avg}) pair, we determine the corresponding 2-dimensional interval, and then look up its $P(R|t)$ value from the training set. However, this method needs large amounts of training data to avoid overfitting. Moreover, it does not give us any insights into the relationship between (P_{ml}, P_{avg}) and $P(R|t)$.

Application of probabilistic classification methods: This approach has been investigated already in [Fuhr

[2]In order to derive a meaningful statistics, elements with less than 100 words were not considered here.

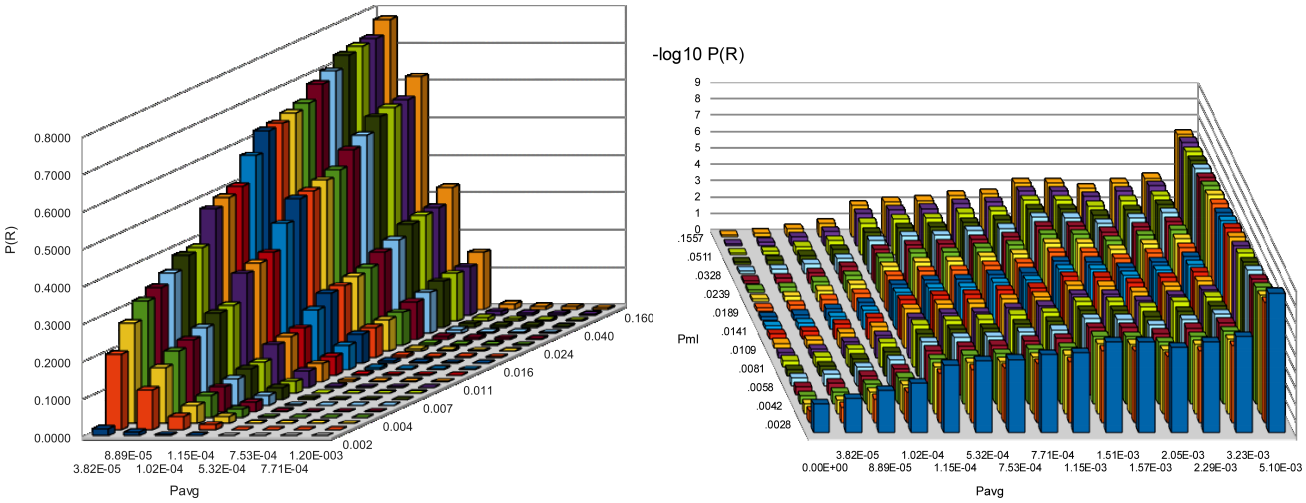


Figure 2: $P(R|t)$ for different (P_{ml}, P_{avg}) values (INEX), linear/log scale

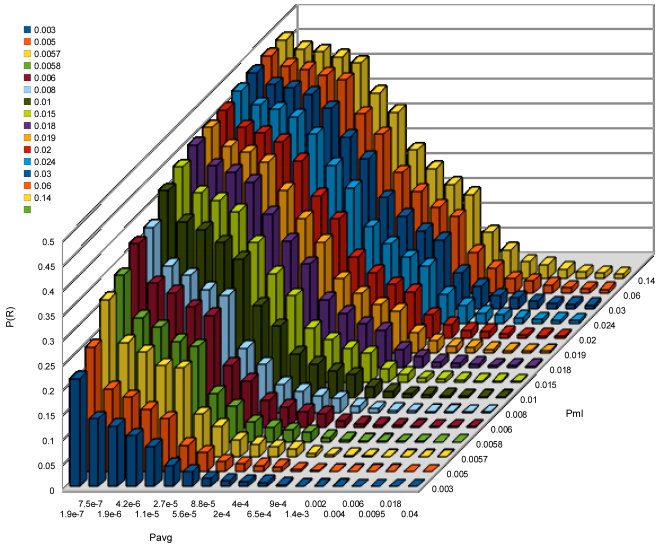


Figure 3: $P(R|t)$ for different (P_{ml}, P_{avg}) values (TREC)

and Buckley, 1991]. As input, the machine learning method would use the raw data underlying the figures from above, i.e., for each term in each query-document pair considered, we have a training instance consisting of the P_{ml} and P_{avg} values as features and the relevance decision as class variable. In recent years, this kind of approach has also become very popular for developing retrieval functions in the so called 'learning to rank' approaches (see e.g. [Fuhr, 1989; Liu, 2009]). Like the previous method, however, this approach operates like a black box, giving us no further insights.

Application of numeric prediction: Here we start with the data shown in figures 2 - 3, and now seek for a function that describes the relationship between P_{ml}, P_{avg} and $P(R|t)$. As classic smoothing functions perform the same kind of task, we can compare the outcome of the machine learning method with these functions.

From these three possibilities, we only consider the last one in the following. Furthermore, we only regard the most simple variant of numeric prediction, namely linear regression.

5 Linear regression

First, we use a purely linear function of the form:

$$\mathrm{P}_s(t_i|d) = \alpha P_{ml} + \beta P_{avg} + \gamma \quad (3)$$

As a second variant, we start from the observation in figure 2 that a linear function of P_{avg} may not be very appropriate. Therefore we use $\log(P_{avg})$ instead:

$$\mathrm{P}_s(t_i|d) = \alpha P_{ml} + \beta \log(P_{avg}) + \gamma \quad (4)$$

Table 1 shows the actual coefficients which have been predicted using linear regression, along with the average squared error. As we can see, replacing P_{avg} by its logarithm (LR linear vs. LR log) reduces the error substantially for both collections.

For further analysis, we regard the difference between the linear predictions of equation 3 and the actual $P(R|t)$ values, as illustrated in figure 4 for the INEX collection (for TREC, the figure looks very similar). In the ideal case, there would be random errors; instead, these figures show

Table 1: Coefficients derived using linear regression

Method	Collection	α	β	δ	γ	Error
LR linear	INEX	0.97	-60.43		0.12	0.053
LR log	INEX	-9.12	-2		9.7	0.011
LR quadratic	INEX	0.97	-209.58	41064.69	0.18	0.022
LR linear cnst.=0	INEX	2.59	-23.4		0	0.060
LR linear	TREC	1.07	-6.93		0.13	0.091
LR log	TREC	-6.23	-0.5		3.43	0.012
LR quadratic	TREC	1.07	-28.03	660.81	0.16	0.041
LR linear cnst.=0	TREC	2.65	-2.69		0	0.094

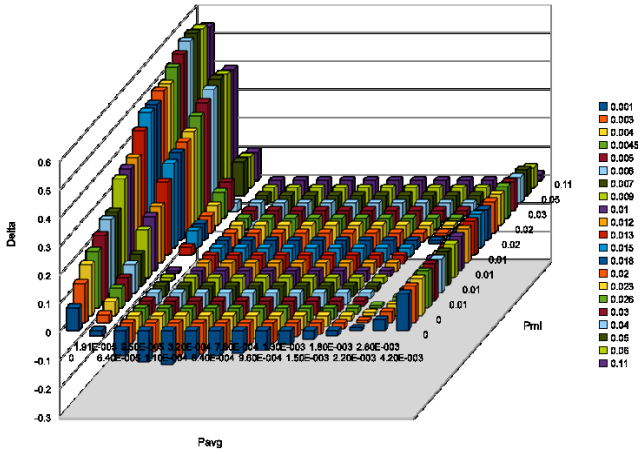


Figure 4: Residuals for linear regression (INEX)

us systematic deviations from the predicted values. The distribution of these errors suggests that a quadratic function of P_{avg} would be more appropriate:

$$\mathrm{P}_s(t_i|d) = \alpha P_{ml} + \beta P_{avg} + \delta P_{avg}^2 + \gamma \qquad (5)$$

Looking at the corresponding quadratic errors in table 1 (LR quadratic), we see that the quadratic form is better than the linear one, and rates as good as the variant with $\log(P_{avg})$.

Since JM smoothing also uses a linear function with P_{ml} and P_{avg} as inputs, we want to compare its outcome with that of our linear regression. For that, we used the equation 2 with $\lambda = 0.7$ which gave the best results for this method. For better comparison, we also tried a variant of the regression function 3, where we dropped the constant γ, so in this case it has the same structure as the JM smoothing function. However, looking at the corresponding regression coefficients listed in table 1 (LR linear cnst=0), we see that P_{avg} has a negative coefficient, whereas JM smoothing assumes both coefficients to be positive. In JM smoothing, P_{ml} is the dominating factor (although P_{avg} has a higher weight with $\lambda = 0.7$, it is at least an order of magnitude smaller than P_{ml}), whereas the empirical data as well as the result of our regression put major emphasis on P_{avg}, and P_{ml} just serves as a minor correction factor.

6 Retrieval experiments

Finally, we performed retrieval experiments with the retrieval function 1 and the various regression functions and compared them with standard retrieval functions. The results are depicted in table 2 and figure 5.

For the three variants of linear regression, we did not separate between training and test sample, so their results are a bit optimistic. Only for the purely linear form, we performed experiments with 2-fold cross validation (LR linear (cv)), showing that the choice of the training sample has little effect on the quality of results.

Comparing the results of the three variants of linear regression, we can see that for both collections, already the linear form gives good results, which can be improved by using one of the variants. For INEX, $\log(P_{avg})$ gives the best quality overall, whereas the quadratic form yields improvements for the top ranking elements only. With TREC, both the logarithmic and the quadratic form are much better than the linear one. In both cases, the quality of JM smoothing is comparable to that of the linear form. BM25 performs poorly for INEX, but very good for TREC.

Furthermore, we also present results for our odds-like language model presented in [Abdulmutalib and Fuhr, 2008], where the retrieval function is shown in eqn. (6); as estimate of $P(d)/P(\bar{d})$, we use the ratio of the length of d and the average document length, and ω and γ are tuning parameters for smoothing.

$$\rho_{o,e}(q,d) = \prod_{t_i \in q^T \cap d^T} \left(\frac{P_{ml}(t_i|d)}{P_{avg}(t_i|C)} \right)^{\omega} \cdot \prod_{t_i \in q^T - d^T} P_{avg}(t_i|C)^{\gamma} \cdot \frac{P(d)}{P(\bar{d})} \qquad (6)$$

Table 2: Retrieval results: empirical smoothing vs. standard retrieval methods (INEX / TREC)

Method	MAP	P@5	P@10	P@20
LR linear	0.0729	0.355	0.339	0.334
LR log	0.1004	0.397	0.366	0.315
LR quadratic	0.0668	0.389	0.389	0.359
JM	0.0667	0.303	0.245	0.216
LR linear (cv)	0.0862	0.331	0.324	0.299
Odds	0.0800	0.348	0.348	0.323
ZL	0.0780	0.338	0.324	0.307
BM25	0.0063	0.096	0.087	0.070

Method	MAP	P@5	P@10	P@20
LR linear	0.0286	0.283	0.253	0.213
LR log	0.0633	0.359	0.312	0.273
LR quadratic	0.0654	0.304	0.247	0.222
JM	0.0307	0.214	0.238	0.231
LR linear (cv)	0.0355	0.345	0.339	0.333
Odds	0.0572	0.232	0.211	0.191
ZL	0.0611	0.279	0.233	0.228
BM25	0.0844	0.445	0.432	0.352

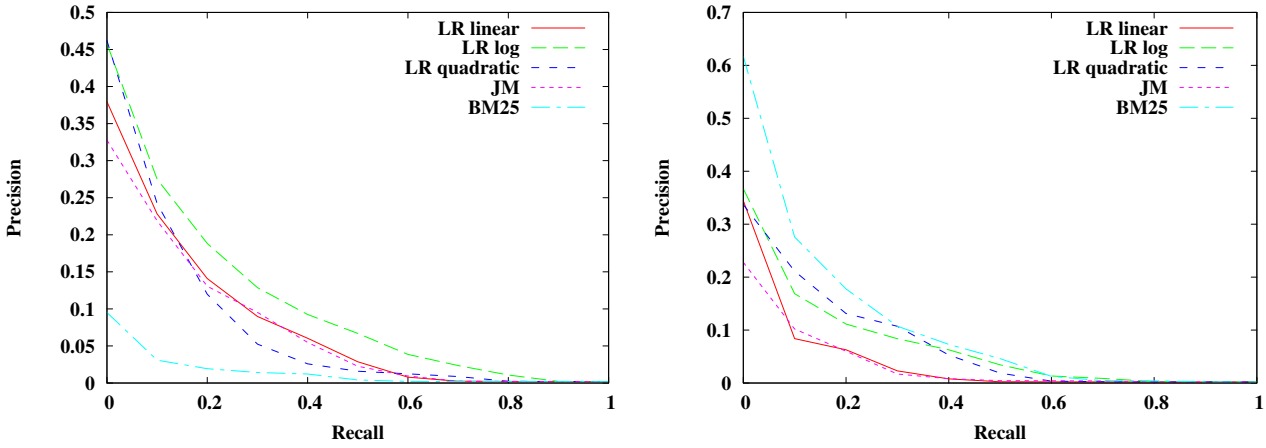


Figure 5: Recall-precision graphs for various smoothing methods and BM25 (INEX / TREC)

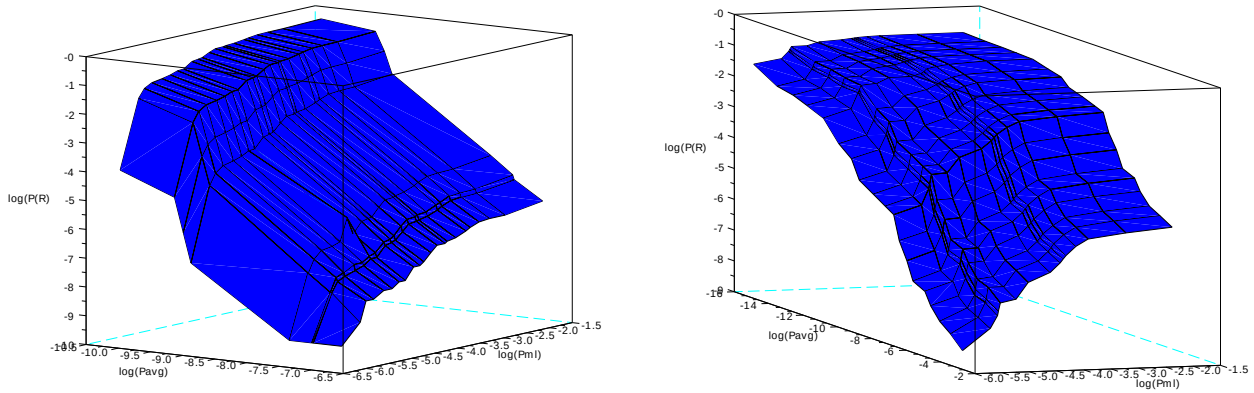


Figure 6: $\log(P(R|t))$ as function of $\log(P_{ml})$ and $\log(P_{avg})$) (INEX / TREC)

Overall, these results show that empirical smoothing in combination with nonlinear regression functions is superior to classic smoothing methods.

7 Further analysis

Figures 2–3 illustrating the relationship between P_{ml}, P_{avg} and $P(R|t)$ do not plot the first two dimensions in proportion to their size. In contrast, figure 6 uses a logarithmic scale for all three dimensions and also plots them proportionally. These figures indicate that the relationship could be fairly well described by a plane in the log-log-log space. In fact, looking at the odds-like-retrieval function (6), this is exactly the form that would result from such a plane (modulo the document length component). Based on this function, we also performed a few experiments with logistic regression, but the results were inferior to that of a grid search for the parameters ω and γ [Abdulmutalib, 2010, sec. 7.8].

8 Conclusion and Outlook

In this paper, we have investigated the relationship between smoothing in language models and idf weights. Although the relative collection frequency P_{avg} and idf weights are very similar, there is a contradiction in the weighting strategy. Regarding the correlation between the language model parameters and relevance, we find that the idf type of weighting seems to be more appropriate. Based on the observed correlation, we have devised empirical smoothing as a new type of term weighting for language models, and retrieval experiments confirm the general applicability of our method. Finally, we showed that the most appropriate form of describing the relationship between the language model parameters and relevance seems to be a product form, which confirms a language model proposed by us before.

In this paper, we have not considered the influence of document length. In fact, other smoothing methods like Dirichlet smoothing or absolute discount (see e.g. [Lafferty and Zhai, 2001]) consider this parameter. Thus, empirical smoothing could also be extended to document length as third parameter.

The comparison between theoretic models and empirical data in this paper has brought us interesting observations. However, this comparison does not answer the question why JM smoothing gives fairly reasonable retrieval results, its structure contradicts our empirical findings. A reasonable explanation for this effect remains the subject of further research.

References

[Abdulmutalib and Fuhr, 2008] Najeeb Abdulmutalib and Norbert Fuhr. Language models and smoothing methods for collections with large variation in document length. In A M. Tjoa and R. R. Wagner, editors, *DEXA Workshops*, pages 9–14. IEEE Computer Society, 2008.

[Abdulmutalib, 2010] Najeeb Abdulmutalib. Language models and smoothing methods for information retrieval. PhD thesis (submitted), 2010.

[Berger and Lafferty, 1999] Adam Berger and John Lafferty. Information retrieval as statistical translation. In *Proceedings of the 22nd International Conference on Research and Development in Information Retrieval*, pages 222–229, New York, 1999. ACM.

[Croft and Harper, 1979] W. B. Croft and D. J. Harper. Using probabilistic models of document retrieval without relevance information. *Journal of Documentation*, 35:285–295, 1979.

[Fuhr and Buckley, 1991] Norbert Fuhr and Chris Buckley. A Probabilistic Learning Approach for Document Indexing. *ACM Trancactions on Information Systems*, 9(3):223–248, 1991.

[Fuhr, 1989] N. Fuhr. Optimum polynomial retrieval functions based on the probability ranking principle. *ACM Transactions on Information Systems*, 7(3):183–204, 1989.

[Hiemstra, 1998] Djoerd Hiemstra. A linguistically motivated probabilistic model of information retrieval. In *Lecture Notes In Computer Science - Research and Advanced Technology for Digital Libraries - Proceedings of the second European Conference on Research and Advanced Technology for Digital Libraries: ECDL'98*, pages 569–584. Springer Verlag, 1998.

[Jelinek and Mercer, 1980] Frederick Jelinek and Robert L. Mercer. Interpolated estimation of markov source parameters from sparse data. In *Proceedings of the Workshop on Pattern Recognition in Practice*, 1980.

[Lafferty and Zhai, 2001] J. Lafferty and C. Zhai. Probabilistic ir models based on document and query generation. In B. Croft J. Callan and J. Lafferty, editors, *Proceedings of workshop on Language Modeling and Information Retrieval*, 2001.

[Liu, 2009] Tie-Yan Liu. Learning to rank for information retrieval. *Found. Trends Inf. Retr.*, 3(3):225–331, 2009.

[Ponte and Croft, 1998] Jay M. Ponte and W. Bruce Croft. A language modeling approach to information retrieval. In *SIGIR'98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 275–281, New York, NY, USA, 1998. ACM Press.

[Roelleke and Wang, 2008] Thomas Roelleke and Jun Wang. Tf-idf uncovered: a study of theories and probabilities. In *SIGIR*, pages 435–442, 2008.

[Zhai and Lafferty, 2001] Chengxiang Zhai and John Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In W. B. Croft, D. Harper, D. H. Kraft, and J. Zobel, editors, *Proceedings of the 24th Annual International Conference on Research and development in Information Retrieval*, New York, 2001. ACM.

An Attribute-based Model for Semantic Retrieval

Hany Azzam, and Thomas Roelleke
School of Electronic Engineering & Computer Science
Queen Mary University of London
London, UK
{hany,thor}@dcs.qmul.ac.uk

Abstract

This paper introduces a knowledge-oriented approach for modelling semantic search. The modelling approach represents both semantic and textual data in one unifying framework, referred to as the probabilistic object-relational content modelling framework. The framework facilitates the transformation of "term-only" retrieval models into "semantic-aware" retrieval models that consist of semantic propositions, such as relationships and classification of objects. To illustrate this facility, an attribute-based retrieval model, referred to as TF-IEF-AF-IDF, is instantiated using the modelling framework. The effectiveness of the developed retrieval model is demonstrated using the Internet Movie Database test collection. Overall, the probabilistic object-relational content model can guide how semantic search and semantic data are modelled.

1 Introduction

Modern retrieval systems have become more complex and semantic-aware by exploiting more than *just* the text, e.g. [Bast *et al.*, 2007; Kasneci *et al.*, 2008]. Nowadays, large-scale knowledge bases can be automatically generated relatively easily from knowledge sources such as Wikipedia or other semantically explicit data repositories such as ontologies and taxonomies that explain entities (e.g. mark-up of persons, movies and locations) and record relationships (e.g. bornIn and actedIn).

Obstacles arise, however, when developing ranking functions and, in a broader sense, search strategies that combine query and document text with other types of evidence derived from semantic-rich knowledge bases. In particular, it is challenging when the ranking function is implemented directly on top of a standard physical document representation, as in the standard information retrieval (IR) engineering approach [Cornacchia and de Vries, 2007]. Consequently, an alternative approach, or more ambitiously, an alternative standard, is required to reduce the complexity of building and maintaining information systems and re-using their retrieval strategies [Fuhr, 1999; Hiemstra and Mihajlovic, 2010]

Design re-use is particularly important since IR already has a well-established family of retrieval models, namely TF-IDF, BM25 and language modelling (LM) that are used in many tasks but in slightly different ways. Ideally, these standard retrieval models should be re-used and adapted to solve complex and semantic retrieval tasks, and overall, to maximise the benefit gained from the underlying data.

How to link the the world of retrieval models with the world of semantic data, while avoiding an extensive engineering process, is not a straightforward process. However, the first step towards transferring the achievements of text retrieval models and so maximising the impact of semantic data, will be to simplify the process of tailoring search to a specific work task [Hawking, 2004].

This paper revisits a framework that may help to establish a standard for developing semantic retrieval models. The framework, referred to as the probabilistic object-relational content model (PORCM, [Roelleke, 1999]), is closely linked to engineering initiatives espoused in the development of business automation solutions using relational database management systems: design a conceptual schema; express the user application in terms of this schema; and design the user interface.

Such a framework, thus, gives creative freedom to designers to invent and refine semantic-aware retrieval models. It also allows for *more than one* semantic-based retrieval model and avoids the need to propose a single retrieval model for semantic search. As such, the framework provides a platform for developing effective semantic retrieval models applicable to textual and semantic data.

1.1 Contributions & Structure

The main contribution of this paper is to use a modelling framework to demonstrate a semantic variant of a standard retrieval model. The framework acts as a logical layer, which decouples the retrieval models from the physical representation of the data (document structure and content), bringing what the database field calls "data independence" to IR systems.

A BM25 motivated semantic retrieval model is instantiated using the probabilistic object-relational content model. This particular model exemplifies how, by taking a knowledge-oriented approach, retrieval models which are traditionally designed for *terms*, i.e. for keyword-based retrieval, can exploit terms and semantic evidence while ensuring data independence.

The feasibility of developing workable retrieval models for semantic search and the effectiveness of the developed retrieval model is demonstrated on the Internet Movies Database (IMDB) collection.

The remainder of this paper is organised as follows. Section 2 highlights some similarities and differences between structured document retrieval and semantic retrieval. The section also outlines related work in the literature. Section 3 details the probabilistic ORCM and its components. Section 4 showcases a retrieval model for semantic retrieval based on the proposed framework. Section 5 evaluates the retrieval model, and Section 6 concludes the discussion.

	Structured Document Retrieval (SDR)	Semantic Retrieval (SR)
Queries	Keyword-oriented, with structural components (e.g. XPATH with "contains" predicate)	Knowledge-based, with keyword-based components
Retrieval Unit	Documents, Sections, etc: structural objects	Documents, Actors, etc: any object
Evidence Spaces	Terms and Element Types (e.g. section, title)	Terms, Class names, Classifications, Relationship names, Relationships, Attribute names, attributes

Figure 1: Structured Document Retrieval versus Semantic Retrieval

2 Background and Related Work

2.1 Structured Document Retrieval & Semantic Retrieval

Figure 1 illustrates some characteristics of structured document retrieval (SDR) and semantic retrieval.

SDR is not limited to a particular structural markup language or a particular forum and so has general applicability. Our discussion, however, focuses on XML as the structural markup language because it is the most widely used standard. Additionally, we focus on INEX-related work and contributions as INEX is characterised by research about SDR, and it is the initiative for the evaluation of XML retrieval [Fuhr *et al.*, 2002].

The first point of comparison is the type/formulation of queries. In SDR the user information need can be formulated using either text-only or text-and-structure approaches. The text-only approach is keyword-based (e.g., the content-only approach in INEX [Amer-Yahia and Lalmas, 2006]). The text-and-structure approach combines textual and structural clues (e.g. XPath, NEXI [Trotman and Sigurbjörnsson, 2004]). Queries in semantic retrieval can be expressed using the aforementioned approaches in addition to the semantic structures found in the formulated query and/or the document representation (e.g. semantic content-and-structure approach [van Zwol and van Loosbroek, 2007]). Semantic queries can also be expressed using graph-patterns such as SPARQL [Prud'hommeaux and Seaborne, 2006]. Extending SPARQL for full-text and semantic search (e.g. [Bast *et al.*, 2007; Kasneci *et al.*, 2008; Elbassuoni *et al.*, 2009]) is analogous to the prior work on SDR which has enhanced XPath and XQuery by various forms of text-search and ranking capabilities. The graph-based approaches, in contrast to XML trees, are independent of the physical representation of the underlying data. If a query is expressed via XQuery, the application would have to know the particular data representation.

The second aspect illustrated in Figure 1 is the retrieval unit (answer type). In SDR the document structure is explicit and, therefore, the retrieval unit is based on the document's presentation and logical structures, such as chapter and section. Semantic retrieval focuses instead on retrieving *objects* which have a particular meaning. This is particularly evident in the discussion of query formulations when semantic text-and-structure topics are introduced to conduct *semantic* retrieval [van Zwol and van Loosbroek, 2007]. Therefore, in semantic retrieval, the answer type has a more general and semantic form, which includes objects such as a person, a product, or a project.

The third difference is the evidence (ranking criteria) used in retrieval. Unlike in traditional IR, in SDR the additional structural evidence not seen in unstructured (flat) text documents is exploited. For example, in XML retrieval the logical document structure is used to estimate the relevance of an element according to the evidence associated with this element only. When the goal is to rank documents, the probabilities of estimating the relevance of each element are combined to produce a single probability for the document. There are two strands of models for element-based ranking and evidence combination: variants of probabilistic models based on the probability ranking principle, such as [Robertson *et al.*, 2004; Lu *et al.*, 2005]; and variants of the statistical language modelling technique proposed by [Ponte and Croft, 1998], such as [Ogilvie and Callan, 2002].

In semantic retrieval, structural elements can be also utilised to estimate the relevance of an object. However, these elements bear a semantic meaning (e.g. actor, director Figure 2) and, thus, a distinctive term distribution. For example, [Kim *et al.*, 2009] extend [Ogilvie and Callan, 2002] to demonstrate how varying weights for different semantic elements across query terms can improve retrieval performance. Evidence associated with semantic annotations in the form of linguistic structures is also used in semantic retrieval, especially in question answering applications. [Zhao and Callan, 2008; Bilotti *et al.*, 2007] propose ranking answer-bearing sentences to questions by incorporating the semantic annotations in both the sentences and queries into the retrieval process.

Semantic annotations expressed in the form of Resource Description Framework (RDF) graphs, referred to as entity-relationship graphs, are also used as ranking criteria. These graphs are used as a source of evidence to construct graph-based ranking models and queries for retrieving semantic objects. For example, [Kasneci *et al.*, 2008; Elbassuoni *et al.*, 2009] propose LM variants for ranking the results of keyword-augmented graph-pattern queries over entity-relationship graphs.

```
<Movies>
  <movie id="329191" >
    <title> Gladiator </title>
    <yeart> 2000 </year>
    <actors><actor id="russell_crowe">Russell Crowe </actor> </actors>
    <team>  <director id="ridley_scott"> Ridley Scott </director> </team>
    ...
    <plot> Maximus is a powerful Roman general ... </plot>
  </movie>
</Movies>
```

Figure 2: XML-based Representation of a Movie

In this paper we utilise full-text and semantic evidence for semantic retrieval. These types of evidence are represented by a set of propositions: terms, classifications, relationships and attributes. For example, in Figure 2 the XML-based representation of a movie contains several types of evidence. If explicated according to the aforementioned set of propositions, then the term proposition would represent the full-text evidence, which is similar to the common keyword-based IR representation. The classi-

fication proposition would capture the "class of' relationship between objects and classes (e.g. director "Ridley Scott"). The relationship proposition would associate two objects, and the attribute proposition would contain the relationship between an object and an atomic value.

The aforementioned propositions stem from object-oriented and content modelling. They support the retrieval of structural elements, semantic elements and heterogeneous objects. The information about the structure and the specifics of objects is represented in the unifying framework of text (terms) and object-oriented modelling (Section 3 discusses the framework in more details).

2.2 Related Work

Related work can be found primarily in investigations of the XML retrieval task, which has been addressed from both IR and database perspectives. The proposed semantic retrieval model is akin to XML retrieval models such as BM25f [Robertson *et al.*, 2004] and hierarchical language modelling [Ogilvie and Callan, 2003] in that it is based on combining different evidence spaces and probabilities.

Our retrieval model is also similar to the models in [Kasneci *et al.*, 2008; Elbassuoni *et al.*, 2009] in the sense that these models focus on retrieving semantic data. However, these approaches mainly propose a variant of only one traditional retrieval model to answer semantic queries; moreover, they do not combine other sources of evidence such as full-text and/or structural elements. On systems-side, approaches such as ESTER [Bast *et al.*, 2007] support a similar class of semantic queries as the proposed model; however, efficiency is their primary design goal.

Our knowledge representation approach shares some aspects with logical approaches for modelling IR such as MIRTL (Multimedia Information Retrieval Terminological Logic, [Meghini *et al.*, 1993]) where "terms" are used to represent concepts and roles. In our framework, however, content is considered separate from the concepts of object-oriented modelling. To implement the retrieval model designed using the proposed knowledge representation, an integrated database and IR approach is used. This approach is similar to probabilistic database approaches found in [Dalvi and Suciu, 2004; Chaudhuri *et al.*, 2006; Roelleke *et al.*, 2008].

3 Knowledge Representation

This section proposes a schema component for semantic retrieval. The notion of a "schema" highlights the difference between keyword-based and semantic retrieval where the former requires a search over *only* an inverted file structure, while the latter requires several processing steps and different representations. The proposed schema is based on object-relational modelling principles. Traditionally, the object-relational model (e.g. [Stonebraker *et al.*, 1998]) uses relations such as "memberOf(...)", "relationship(...)" and "attribute(...)" to model concepts such as classification, relationships, and attributes. We extend this approach and introduce an object-relational content model. The model integrates object-relational modelling and content-oriented (term-based) modelling into one framework. Consequently, we extend the model to create its probabilistic variant, namely the probabilistic object-relational content model. This model includes relations which represent probabilistic parameters to model IR-like retrieval models.

3.1 Object Relational Content Model (ORCM)

Figure 3 uses the ORCM to represent the movie in Figure 2. Ellipses indicate that some data have been omitted to conserve space. The location where different elements occur are stored as paths, expressed in XPath. For readability we use a simplified syntax, e.g., "imdb/movie_1/title_1" points to the attribute describing a movie's title.

term	
Term	Context
gladiator	329191/title[1]
2000	329191/year[1]
russell	329191/.../actor[1]
crowe	329191/.../actor[1]
ridley	329191/.../director[1]
scott	329191/.../director[1]
maximus	329191/plot[1]
powerful	329191/plot[1]
roman	329191/plot[1]
general	329191/plot[1]
...	...

term_doc	
Term	Context
gladiator	329191
2000	329191
russell	329191
crowe	329191
ridley	329191
scott	329191
maximus	329191
powerful	329191
roman	329191
general	329191
...	...

(a) Term propositions in the element and root contexts

classification		
ClassName	Object	Context
movie	329191	movies[1]
title	329191/title[1]	329191
year	329191/year[1]	329191
actors	329191/actors[1]	329191
actor	329191/.../actor[1]	329191/actors[1]
team	329191/team[1]	329191
director	329191/.../director[1]	329191/team[1]
plot	329191/plot[1]	329191
...	...	...

(b) Classification propositions

attribute			
AttrName	Object	Vatlue	Context
id	329191	"329191"	movies[1]
id	329191/.../actor[1]	"russel..."	32.../actors[1]
id	329191/.../director[1]	"ridley..."	32.../team[1]
...	...	...	...

(c) Attribute propositions

Figure 3: An Object-Relational Content Model Representing a Movie

Traditionally, in order to model the task of document retrieval, a term-document representation based on a data structure such as "term(Term, DocId)" would suffice. For example, in Figure 2 the terms in the XML fragment can have a flat representation (the XML elements are not interpreted), such as "term(movie,329171)". Consequently, for document retrieval the retrieval models (e.g., TF-IDF, language modelling, BM25) for ranking documents are primarily based on a "term(Term,DocId)" relation.

In the case of SDR, a structural representation, such as "term(Term, SecId)", is necessary. This is because the contexts or the document structure (e.g. abstract, section, paragraph) are explicit. Additionally, the retrieval models are usually based on combining the scores obtained from scoring every context, or combining the term frequencies. What then, are the data structures which we use to model (implement) semantic retrieval?

We first review the design process of the ORCM [Roelleke, 1999] and then demonstrate how it can be utilised for semantic retrieval.

The ORCM combines object-oriented and content-based modelling concepts. The object-oriented concepts include classification, relationships and attributes, which are more generally referred to as propositions – a specification stemming from object-oriented modelling and terminological logics [Meghini *et al.*, 1993]. Content modelling is analogous to the traditional IR representation of text, which is usually a term-context-based representation. However, unlike the conventional modelling approaches for IR, such as terminological logic, [Meghini *et al.*, 1993]), "content" is viewed as separate from the concepts of object-oriented modelling. This separation helps content to be described in a more formal and knowledge-oriented way.

There are two design steps taken in order to achieve this separation. Each predicate within a proposition is associated with a context[1] and a term proposition is used as the keyword-based IR representation for text.

Other propositions (components) that can be taken into account include generalisation and aggregation, where generalisation is a relationship between classes, and aggregation is a particular relationship between objects (entities). However, it is the four aforementioned propositions with which we shall be mainly concerned here. The pillars of the probabilistic object-relational content model can be summaries as follows:

- classification of objects: monadic predicate of the form "ClassName(Object)", for example: "actor(russell_crowe)".
- relationship between objects: dyadic predicate of the form "RelationshipName(Subject,Object)", for example: "directedBy(329191,ridley_scott)".
- attribute of objects: dyadic predicate of the form "AttributeName(Object, Value)", for example: ridley_scott.name("Ridley Scott").

In order to implement the object-oriented modelling concepts, a relational approach is used, resulting in the proposed object-relational schema. Relations such as "classification" and "relationship" are devised. Moreover, in order to join object-oriented modelling with keyword-based and content-oriented modelling, an additional predicate, namely "term", is used, and an additional attribute column, namely "Context" is adjoined to term, classification, relationships and attributes. This yields the ORCM modelling paradigm.

Below we contrast the conventional object-relational model with the object-relational content model[2].

Object-relational modelling (ORM):

- classification(ClassName, Object)
- relship(RelshipName, Subject, Object)
- attr(AttrName, Object, Value)

Object-relational content modelling (ORCM):

- classification(ClassName, Object, *Context*)
- relship(RelshipName, Subject, Object, *Context*)
- attr(AttrName, Object, Value, *Context*)
- *term(Term, Context)*

[1]Context is a general concept that refers to documents, sections, databases or any other object with a content

[2]In the schema design process we often opt to use shorter names for relation and attribute names so that to achieve a slimmer form of the schema.

Terms are complementary to classification and relationships. Most importantly, and one of the main emphases of this modelling paradigm, is that content is not modelled, for example, as a relationship "contains(DocumentId, Term)", but rather content is modelled by maintaining an attribute column "Context" in the schema for each proposition. In other words, content is modelled separately from existing concepts, such as classification and relationships.

There are several fundamental benefits of utilising the object-relational content model. One benefit is that knowledge modelling, in general, aids "knowledge architects" to build information and knowledge management systems that are both flexible and scalable. Another benefit is that it facilitates the transformation of term-document-based IR retrieval models into retrieval models founded on the probabilistic object-relational content model, thus resulting in a strand of retrieval models suited for semantic retrieval. Lastly, the model enables the representation of textual, structural and semantic data in one unifying framework. The uniform representation of the data, the semantic retrieval models and the decoupling between the two using the object-relational content model results in data independence. This is a desirable feature when designing complex retrieval systems.

In summary, the ORCM is to be understood as a conceptual model with a set of relations – a relation for each basic concept of object-oriented and content-based modelling.

3.2 Probabilistic Spaces for Semantic Retrieval

The evidence space (ranking criteria) construction is facilitated by the probabilistic object relational content model. The probabilistic ORCM comprises the relations of the ORCM, as well as relations representing probabilistic parameters. For example, for the basic relation "term_doc(Term,Doc)", there can be term-based and document-based probabilities.

Some of the probabilistic relations for "term_doc" are:

p_DF_t_term_doc(T): Document frequency-based probability of term t derived from relation "term_doc(Term,Doc)"

p_TF_t_term_doc(T): Tuple frequency-based probability of term t derived from relation "term_doc(Term,Doc)"

pidf_term_doc(T): Inverse document frequency (IDF)-based probability of term t derived from relation "term_doc(Term,Doc)"

In probabilistic ORCM the techniques and models of IR devised for term-based retrieval models become available for class-based, relationship-based and attribute-based retrieval. Concepts such as the tuple frequency-based probability of a class (class-frequency, *CF*) and the IDF of a class name (similar to the IDF of a term) make immediate sense. Similarly, the tuple frequency-based probability of an attribute name (attribute name-frequency, *AF*) and the IDF of an attribute name become possible.

The ability to transfer the achievements of term-based retrieval directly to semantic retrieval models makes the probabilistic ORCM a potential base for semantic retrieval. Moreover, the way probabilistic spaces can be combined in probabilistic ORCM can lead to new and effective retrieval models. For example, the frequencies of attributes names are exploited to define an attribute-based retrieval status value (RSV), and this RSV can be combined with other RSV's, such as the term-based one.

The next section provides an example of a retrieval model constructed on top of the discussed representation.

4 An Attribute-based Retrieval Model

The proposed model, TF-IEF-AF-IDF, for semantic retrieval focuses on combining evidences from attribute name and term predicate spaces. The model is implemented in three phases. Figure 4 (Page 6) is a snapshot of the proposed model's components when answering query number 28, "gladiator action maximus scott", from the IMDB test collection. We detail the three phases below.

Phase 1 retrieves "Term-Document's Element-Query" triplets for each query term. Such retrieval can be performed using any term-based retrieval model (e.g. TF-IEF, TF-IDF, BM25, LM). We choose here TF-IEF, where TF is the within-element term frequency, and IEF is the inverse element frequency of a term. TF-IEF is defined as follows:

Definition 1 *TF-IEF:*

$$\mathrm{RSV}_{\textit{TF-IEF}}(e,q) := \sum_{t \in e \cap q} \mathrm{TF}(t,e) \cdot \mathrm{IEF}(t) \qquad (1)$$

"t" stands for term, "e" for element, such as "title", and IEF is inverse element frequency.

Phase 2 consists of two parts. The first part infers the attribute name and root context (root nodes) from the document's elements in the "Term-Document's Element-Query" triplet. This yields an intermediate and query-dependent attribute-based index (this index corresponds to the tf_ief_match_augmented in Figure 4).

The second part infers for each query term its top-k corresponding "context type". For example, for a query such as "fight brad pitt" the inferred top–1 context type would be "title" for query term "fight" and "actor" for query terms "brad" and "pitt". This is because "fight" occurs in the context of type "title", and "brad" and "pitt" occur in the context of type"actor". The attribute-based index constructed in the first part of Phase 2 is used to infer the mapping between each query term and its type. The result of this inference is represented in Figure 4 ("AttrName" corresponds to the context type in which the query term occurs).

qTermAttr(Term, AttrName, Query)

The probability of the mapping between a query term and an attribute type is estimated using the number of mappings between a term and an attribute name divided by the total number of mappings in the intermediate index. The intuition behind the mapping is that if a term occurs frequently within a certain context type then the term is more likely to be "characterised" by that particular context [Kim *et al.*, 2009].

Phase 3 combines an attribute-based retrieval score with a traditional topical (term-based) score resulting in the TF-IEF-AF-IDF model. The motivation to do so is that in some cases an attribute-only retrieval score is deemed unsuitable for estimating the relevance of a particular semantic object with respect to a query. In other words, not all queries are issued with a particular semantic predicate or relationship in mind which the query terms can be mapped to. Therefore, a document-based retrieval score would provide a more realistic setting whereby both the attribute-based and document-based retrieval scores are considered. The intuition behind this combination is comparable to the mixture of document-based and element-based language model scores [Zhao and Callan, 2008].

We formally define the model and its components below.

Definition 2 *The TF-IEF-AF-IDF Model: TF-IEF-AF-IDF model is a multi-stage retrieval model. In the first stage, TF-IEF is used to associate for each query term the document elements and query. Also, each query term is associated with an attribute name. Then, the RSV's of attribute-based retrieval and term-based retrieval are combined into an overall score.*

Let d' be the document inferred from d, where the inference assigns several attribute names (context types) to each "Term-Document's Element-Query" generated in phase 1.

Let q' be the query inferred from q, where the inference assigns several attribute names (context types) to each query term.

$$\begin{aligned}\mathrm{RSV}_{\textit{TF-IEF-AF-IDF}}(d,q) := \\ \mathrm{RSV}_{\textit{TF-IDF}}(d,q,\textit{term-based-index-all-docs}) + \\ \mathrm{RSV}_{\textit{AF-IDF}}(d',q',\textit{attribute-based-index-retrieved-docs})\end{aligned} \qquad (2)$$

The term-based score is defined as follows:

Definition 3 *TF-IDF:*

$$\mathrm{RSV}_{\textit{TF-IDF}}(d,q) := \sum_{t \in d \cap q} \mathrm{TF}(t,d) \cdot \mathrm{IDF}(t) \qquad (3)$$

The attribute-based score is defined as follows:

Definition 4 *AF–IDF:*

$$\mathrm{RSV}_{\textit{AF-IDF}}(d,q) := \sum_{a \in d \cap q} \mathrm{AF}(a,d) \cdot \mathrm{IDF}(a) \qquad (4)$$

$\mathrm{AF}(a,d)$ and $\mathrm{TF}(t,d)$ correspond to the within-document *attribute name* frequency and the within-document *term* frequency components, respectively. "aName" is an attribute name, "d" is a document and "q" is the query. The frequencies are estimated using BM25's TF_K(t,d) ($\mathrm{tf}_d/(\mathrm{tf}_d + K_d)$) quantification [Robertson, 2004]. tf_d is total frequency and K is a normalisation factor reflecting the document length. In the attribute-based aggregation it is af_d instead of tf_d and K is the number of attributes in the intermediate index. The IDF in the AF-IDF is calculated over the set of *retrieved* documents only.

Figure 4 demonstrates for the IMDB's query number 28 the processing steps and relational instantiations of the TF-IEF-AF-IDF model (see Section 5 for details on the IMDB collection). "tf_ief_match" represents the retrieved Term-DocumentElement-Query triplets for each query term. "tf_ief_match_augmented" infers attribute names and root contexts for each Term-DocumentElement-Query triplet, and "attr_index" is the representation of the attribute names and root contexts. "qTermAttr" represents the query term and its inferred attribute name. Lastly, "AF-IDF" represents the predicate-based attribute retrieval scores and "TF-IEF-AF-IDF" represents the combined predicate-based attribute and term retrieval scores.

4.1 Discussion

The model described above integrates several models into a stepwise retrieval process. Such a retrieval process is similar to "database matching" which is achieved in several steps, using different representations at each database level. The stepwise approach is more complex than simple flat text matching and creates opportunities for more powerful matching specifications for semantic search.

The framework ensures that traditional models such as TF-IDF and aggregation techniques, such as BM25's TF_K quantification are transferrable to models where semantic knowledge is explicated, such as AF-IDF.

tf_ief_match			
$P(t,e\|q)$	Term	Retrieved_Element	Query
0.774433	gladiator	imdb/movie_128903/title[1]	q028
...	...	...	q028
0.392531	action	imdb/movie_113798/genre[1]	q028
...	...	...	q028
0.426180	maximus	imdb/movie_2112/plot[1]	q028
...	...	...	q028
0.187669	scott	imdb/movie_284995/team[2]	q028

(a) Element-based Retrieval Results

tf_ief_match_augmented: inferred attributes and contexts					
Prob	Term	Retrieved_Element	Attribute_Name	Root_Context	Query
0.77	gladiator	imdb/movie_128902/title[1]	title	imdb/movie_128902	q028
...	...	...	...	...	q028
0.39	action	imdb/movie_113798/genre[1]	genre	imdb/movie_113798	q028
...	...	...	...	...	q028
0.43	maximus	imdb/movie_2112/plot[1]	plot	imdb/movie_2112	q028
...	...	...	...	...	q028
0.19	scott	imdb/movie_284995/team[2]	team	imdb/movie_284995	q028

(b) Augmented Retrieval Result: Inferred Attributes Names and Root Contexts

attr_index(AttrName, Context)			
Prob	Attribute_Name	Context	Query
0.77	title	imdb/movie_128902	q028
...	...	...	q028
0.39	genre	imdb/movie_113798	q028
...	...	...	q028
0.43	plot	imdb/movie_2112	q028
...	...	...	q028
0.19	team	imdb/movie_284995	q028

(c) Attribute-based Index

idf: over attr_index		
P(aName)	Attribute_Name	Query
...	...	q028
0.48	title	q028
0.31	plot	q028
0.14	team	q028
0.09	genre	q028
...	...	q028

(d) IDF of attribute names over attr_index

qTermAttr (IMDB query 28)			
Prob	Term	Attribute_Name	Query
0.000426942	gladiator	title	q028
...	...	...	q028
0.616951	action	genre	q028
...	...	...	q028
0.00000521851	maximus	plot	q028
...	...	...	q028
0.14484	scott	team	q028

(e) Query Representation: Terms and Inferred Attributes

AF_K–IDF	
$Prob$	Context
...	...
0.17	imdb/movie_128902
...	...
0.27	imdb/movie_128908

(f) Attribute-Frequency-based Score

TF-IEF-AF-IDF	
RSV	Context
0.33	imdb/movie_128902
0.32	imdb/movie_128908
...	...
...	...

(g) TF-IEF-AF-IDF Score

Figure 4: TF-IEF-AF-IDF Retrieval Phases (IMDB Query 28)

In particular, the attribute-based aggregation (the AF_K component) is instrumental to the performance of the TF-IEF-AF-IDF retrieval model as will be illustrated in the evaluation section. The query terms and the retrieved elements (Phase 2) are mapped to their corresponding semantic predicates, which, in this case, are semantic attributes names. This mapping, then, results in an aggregation over the attribute names instead of the terms.

Our analysis suggests that this shift, hence, leads to an event space that contains less number of *distinct* events (attribute names) but that occur frequently. Such a feature is well-suited to the BM25-like aggregation of frequencies because if an event occurs in a context then the probability it occurs again is greater than the initial probability, i.e. the occurrence of an event depends on previous occurrences – [Wu and Roelleke, 2009] have proposed a probabilistic semantics for this feature, referred to as "semi-subsumed". The non-linear nature of the aggregation is key to the good retrieval quality of TF-IEF-AF-IDF.

To illustrate the proposed retrieval model we used an XML-based collection. Particular to this collection is that each element type has specific semantics and, thus, a distinctive term distribution. This is analogous to, for example, entity relationship graphs where the semantics of the data is is represented rather than the structural layout.

However, unlike its full-fledged semantic counterpart, the XML data "as it comes" does not explicate relationships between entities. This is reflected in the ORCM representation as there are mainly terms, classification and attribute relations. Furthermore, the "basic" representation of the XML data still uses XPath expressions to denote object Id's and contexts. This can be viewed as problematic since the main aim here is to achieve a semantic as opposed to structural representation and eventually semantic as opposed to structural retrieval models.

There are two main solutions to this problem. The first is that the XML data in itself consists of element types that have specific semantics and therefore differ from logical or layout element types that are concerned with a document's or a page's presentation. Secondly, the structural representation can be lifted to become a semantic representation. The following Datalog rule exemplifies how this can be done. The rule for "actor" underlines that a "semantic" object can be extracted by combining structural information about elements of type actor and their attributes (e.g. "russell_crowe" in Figure 2).

```
actorElement(XPath, Context) :−
    classification (actor, XPath, Context);

actorEntity (ObjectId, Context) :−
        actorElement(XPath, Context) &
         attribute (id, XPath, ObjectId, Context);
```

The above rules derive a semantic Id for an actor; moreover, it lifts the "structural" into a "semantic" classification. In a similar fashion, attributes can be lifted to become relations in "higher-level" layers of the ORCM. These layers can be derived from the "basic" ORCM and form an abstraction hierarchy from basic to structural to semantic schema. This helps to achieve data independence, as any data (XML, RDF, RSS) can be represented in the basic ORCM, and then, application-specific relations are derived.

Overall, this discussion emphasises that the ORCM schema supports reasoning over structural elements, semantic structures and, eventually, semantic information which is "naturally" present in entity relationship graphs.

5 Evaluation

The purpose of the evaluation is two-fold. Firstly, it proves the feasibility and applicability of the proposed knowledge representation, namely the probabilistic object relational model, for both term-based and semantic retrieval. Secondly, it investigates the quality of the proposed retrieval model, which is *one instance* of the proposed knowledge representation.

	MAP	RecipRank
TF_K–IDF	35.07	36.80
TF-IEF-AF-IDF-top-1	*52.11*	*53.96*
TF-IEF-AF-IDF-top-5	**60.32**	**62.04**
Improvement	+71.19	+68.59

Figure 5: Retrieval Performance per Query Mapping (boldface indicate best performing model and results are in percentages)

The experiment was performed on the IMDB collection[3], which consists of $437,281$ documents or XML records. Each document corresponds to a movie and was constructed from text data. The element types were "title", "year", "releasedata", "language", "genre", "country", "location", "colorinfo", "cast", "team" and "plot". Document content consists mostly of keywords, with the exception of the plot element.

We utilised the 40 queries and relevance criteria in [Kim *et al.*, 2009]. For each query, a query term is mapped to its corresponding semantic structure. This leads to a set of queries that contain keywords and semantic predicates (attributes). The unit of retrieval for all queries is the movie object. Below is a logical representation of query number 28 with top–1 mapping.

```
retrieve (X) :−
 X. title ( gladiator ) & X.genre(action ) &
 X.actor(maximus) & X.plot(maximus) & X.director( scott );
```

For the experiments, we used HySpirit [Roelleke *et al.*, 2001], a probabilistic reasoning system which supports the retrieval of text and (semi-)structured data. We chose HySpirit because it provides a framework with high-level and customisable concepts for modelling retrieval models. The framework provides an open-box approach for describing ranking models for any object.

Figure 5 shows the retrieval effectiveness for the test queries on the IMDB collection. To conserve space, only the performance of TF-IEF-AF-IDF with "top–1" and "top–5" mapping has been reported. The main observation is that the proposed method, an attribute-based model for semantic retrieval, significantly outperforms (p-value < 0.01 with two-tailed t-test) the TF_K–IDF baseline. The baseline is a document-oriented retrieval model where the XML elements are discarded (similar to the method reported in [Theobald *et al.*, 2005]).

The improvement performance can be accredited to the combination of term-based and attribute-based evidence spaces. Furthermore, the TF and AF parameters of the model are set to the BM25-like quantification that delivers the best performance since it mitigates the sub-optimal independence assumption of the total count.

The AF component, in particular, reflects that if a term occurs in a document then the probability that it occurs

[3] http://www.imdb.com/interfaces#plain

again is greater than the initial probability, i.e. the occurrence of an event depends on previous occurrences. Switching from term to attribute space, which groups terms under a particular context type, is conducive to retrieval performance.

Overall, the evaluation demonstrates that the expressiveness of the probabilistic ORCM model can lead to an effective model for semantic retrieval.

6 Summary & Conclusion

This paper demonstrates an approach for representing knowledge, how to merge object-relational and term-oriented modelling and how to steer term-based modelling towards semantic modelling (the semantic knowledge is explicit); therefore, it contributes a discussion of how object-relational modelling meets content modelling and its effect on probabilistic retrieval models for semantic retrieval.

Semantic retrieval requires models that, in sound and transparent ways, mix various frequencies and probabilities. Whereas in text retrieval, probabilities of terms are the dominating players, in semantic retrieval, probabilities of terms, classes, relationships, attributes and objects are the parameters involved in the design of a retrieval model. This paper introduces a particular retrieval model, namely TF-IEF-AF-IDF, in which the attribute frequency of retrieved elements (attributes) is a crucial component of ranking retrieved objects (contexts). The AF component gathers evidence from different query terms into one attribute. This aggregation has a positive outcome on retrieval quality, especially when combined with traditional term-based retrieval, as shown in this paper for TF_K–IDF.

The proposed model is one instance of a large family of models that can be developed using the probabilistic object relational content framework. The framework helps to transform retrieval models that are traditionally designed for *terms*, i.e. for keyword-based retrieval to models that are based on *propositions* and, hence, tailored towards more complex and semantic retrieval tasks. Furthermore, the flexibility and openness of the framework encourages engineers to create a variety of retrieval models that combine textual, structural and semantic sources of evidence.

We have contributed to two related facets of semantic retrieval: knowledge representation and retrieval strategy modelling. Future work will investigate other retrieval models for semantic retrieval based on the probabilistic object-relational content model.

7 Acknowledgments

We would like to thank Jinyoung Kim of the University of Massachusetts Amherst for providing us with the collection and the queries. We would also like to thank the reviewers for their excellent suggestions.

References

[Amer-Yahia and Lalmas, 2006] Sihem Amer-Yahia and Mounia Lalmas. Xml search: languages, inex and scoring. *SIGMOD Rec.*, 35(4):16–23, 2006.

[Bast *et al.*, 2007] Holger Bast, Alexandru Chitea, Fabian M. Suchanek, and Ingmar Weber. Ester: efficient search on text, entities, and relations. In *SIGIR*, 2007.

[Bilotti *et al.*, 2007] Matthew W. Bilotti, Paul Ogilvie, Jamie Callan, and Eric Nyberg. Structured retrieval for question answering. In *SIGIR*, pages 351–358, 2007.

[Chaudhuri *et al.*, 2006] Surajit Chaudhuri, Gautam Das, Vagelis Hristidis, and Gerhard Weikum. Probabilistic information retrieval approach for ranking of database query results. *ACM Trans. Database Syst.*, 31(3):1134–1168, 2006.

[Cornacchia and de Vries, 2007] R. Cornacchia and A. P. de Vries. A Parameterised Search System. In *Proceedings of the European Conference on IR Research (ECIR)*, 2007. Best student paper award.

[Dalvi and Suciu, 2004] Nilesh N. Dalvi and Dan Suciu. Efficient query evaluation on probabilistic databases. In *VLDB*, pages 864–875, 2004.

[Elbassuoni *et al.*, 2009] Shady Elbassuoni, Maya Ramanath, Ralf Schenkel, Marcin Sydow, and Gerhard Weikum. Language-model-based ranking for queries on rdf-graphs. In *CIKM '09: Proceeding of the 18th ACM conference on Information and knowledge management*, pages 977–986, New York, NY, USA, 2009. ACM.

[Fuhr *et al.*, 2002] Norbert Fuhr, Norbert Goevert, Gabriella Kazai, and Mounia Lalmas. Inex: Intitiative for the evaluation of xml retrieval. In *ACM SIGIR Workshop on XML and Information Retrieval*, 2002.

[Fuhr, 1999] N. Fuhr. Towards data abstraction in networked information retrieval systems. *Information Processing and Management*, 35(2):101–119, 1999.

[Hawking, 2004] David Hawking. Challenges in enterprise search. In *ADC*, pages 15–24, 2004.

[Hiemstra and Mihajlovic, 2010] Djoerd Hiemstra and Vojkan Mihajlovic. A database approach to information retrieval: The remarkable relationship between language models and region models. Technical Report arXiv:1005.4752, May 2010. Comments: Published as CTIT Technical Report 05-35.

[Kasneci *et al.*, 2008] Gjergji Kasneci, Fabian M. Suchanek, Georgiana Ifrim, Maya Ramanath, and Gerhard Weikum. Naga: Searching and ranking knowledge. In *ICDE*, pages 953–962, 2008.

[Kim *et al.*, 2009] Jinyoung Kim, Xiaobing Xue, and W. Bruce Croft. A probabilistic retrieval model for semistructured data. In *ECIR*, pages 228–239, 2009.

[Lu *et al.*, 2005] Wei Lu, Stephen E. Robertson, and Andrew MacFarlane. Field-weighted xml retrieval based on bm25. In *INEX*, pages 161–171, 2005.

[Meghini *et al.*, 1993] C. Meghini, F. Sebastiani, U. Straccia, and C. Thanos. A model of information retrieval based on a terminological logic. In R. Korfhage, E. Rasmussen, and P. Willett, editors, *Proceedings of the Sixteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 298–308, New York, 1993. ACM.

[Ogilvie and Callan, 2002] Paul Ogilvie and Jamie Callan. Language models and structured document retrieval. In *INEX Workshop*, pages 33–40, 2002.

[Ogilvie and Callan, 2003] P. Ogilvie and J. Callan. Language models and structured document retrieval, 2003.

[Ponte and Croft, 1998] J.M. Ponte and W.B. Croft. A language modeling approach to information retrieval. In W. Bruce Croft, Alistair Moffat, C. J. van Rijsbergen, Ross Wilkinson, and Justin Zobel, editors, *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 275–281, New York, 1998. ACM.

[Prud'hommeaux and Seaborne, 2006] Eric Prud'hommeaux and Andy Seaborne. SPARQL Query Language for RDF. Technical report, W3C, 2006.

[Robertson *et al.*, 2004] Stephen E. Robertson, Hugo Zaragoza, and Michael J. Taylor. Simple BM25 extension to multiple weighted fields. In *CIKM*, pages 42–49, 2004.

[Robertson, 2004] S.E. Robertson. Understanding inverse document frequency: On theoretical arguments for idf. *Journal of Documentation*, 60:503–520, 2004.

[Roelleke *et al.*, 2001] Thomas Roelleke, Ralf Luebeck, and Gabriella Kazai. The HySpirit retrieval platform, demonstration. In Bruce Croft, David J. Harper, Donald H. Kraft, and Justin Zobel, editors, *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, New Orleans, USA*, New York, August 2001. ACM.

[Roelleke *et al.*, 2008] Thomas Roelleke, Hengzhi Wu, Jun Wang, and Hany Azzam. Modelling retrieval models in a probabilistic relational algebra with a new operator: the relational Bayes. *VLDB J.*, 17(1):5–37, 2008.

[Roelleke, 1999] Thomas Roelleke. *POOL: Probabilistic Object-Oriented Logical Representation and Retrieval of Complex Objects*. Shaker Verlag, Aachen, 1999. Dissertation.

[Stonebraker *et al.*, 1998] Michael Stonebraker, Dorothy Moore, and Paul Brown. *Object-Relational DBMSs: Tracking the Next Great Wave*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1998.

[Theobald *et al.*, 2005] Martin Theobald, Ralf Schenkel, and Gerhard Weikum. Topx and xxl at inex 2005. In *INEX*, pages 282–295, 2005.

[Trotman and Sigurbjörnsson, 2004] Andrew Trotman and Börkur Sigurbjörnsson. Narrowed extended xpath i (nexi). In *INEX*, pages 16–40, 2004.

[van Zwol and van Loosbroek, 2007] Roelof van Zwol and Tim van Loosbroek. Effective use of semantic structure in xml retrieval. In *ECIR*, pages 621–628, 2007.

[Wu and Roelleke, 2009] Hengzhi Wu and Thomas Roelleke. Semi-subsumed events: A probabilistic semantics for the BM25 term frequency quantification. In *ICTIR (International Conference on Theory in Information Retrieval)*. Springer, 2009.

[Zhao and Callan, 2008] Le Zhao and Jamie Callan. A generative retrieval model for structured documents. In *CIKM '08: Proceeding of the 17th ACM conference on Information and knowledge management*, pages 1163–1172, New York, NY, USA, 2008. ACM.

Binary Histograms for Resource Selection in Peer-to-Peer Media Retrieval

Daniel Blank and Andreas Henrich
University of Bamberg
D-96052, Bamberg, Germany
{daniel.blank | andreas.henrich}@uni-bamberg.de

Abstract

With the ever increasing amount of media data and collections on the world wide web and on private devices arises a strong need for adequate indexing and search techniques. Trends such as personal media archives, social networks, mobile devices with huge storage space and networks with high bandwidth capacities make distributed solutions and peer-to-peer (P2P) systems attractive. Here, resource selection can be applied to determine a ranking of promising resources based on descriptions of their content. Resources are contacted in ranked order to retrieve appropriate media items w.r.t. a user's information need.

In this paper we apply and adapt resource descriptions in the form of binary histograms and corresponding selection techniques which were designed for low-dimensional spatial data to high-dimensional data in the context of content-based image retrieval (CBIR). W.r.t. related work in distributed information retrieval, which is also discussed in this paper, a main characteristic of our approach are more space efficient resource descriptions. This makes them applicable for a wider range of application fields apart from the P2P domain.

1 Introduction

In recent years, there has been a tremendous increase in (personal) web data. Web users maintain blogs, twitter their lives and upload photos and videos to social media sites. Besides storing media items, people tend to share them with friends and interact with each other by collaboratively tagging or commenting on various items. Consequently, heterogeneous online resources which differ in size, media type and update characteristic have to be administered [Thomas and Hawking, 2009]. Hence, effective and efficient retrieval techniques are essential.

Several criteria can be employed for the retrieval of media items (cf. Fig. 1): *a)* textual content *b)* geographic footprints, *c)* timestamps and *d)* (low-level) audio or visual content information. Based on these criteria, text, image, audio and video documents can be indexed and searched.

Peer-to-peer (P2P) scenarios for the administration of media collections are attractive for multiple reasons. Media items can reside on individual devices without a need to store them on remote servers hosted by service providers.

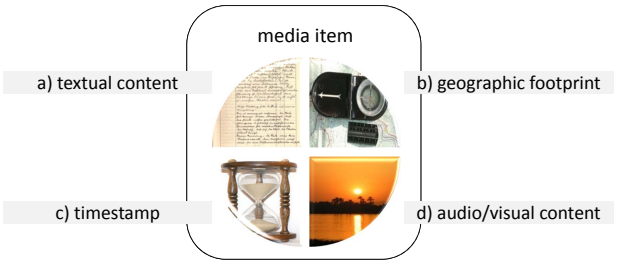


Figure 1: Possible criteria for media retrieval.

Besides reducing dependency from service providers as informational gatekeepers, no expensive infrastructure has to be maintained by applying a scalable P2P protocol such as Rumorama [Müller *et al.*, 2005b]. Crawling, which consumes large amounts of web traffic [Bockting and Hiemstra, 2009], can thus be avoided. Idle computing power in times of inactivity can be used to maintain, analyze and enrich media items.

Our work focuses on space efficient resource description and corresponding selection techniques which allow for efficient and effective query processing. As a proof-of-concept, we design them for the use within Rumorama without limiting their possible application. The resource description and selection techniques can also be applied in the context of traditional distributed information retrieval (IR) (cf. Sect. 2.1) or other variants of P2P IR systems (cf. Sect. 2.2). Furthermore, there is a range of possible application fields apart from P2P IR systems (cf. Sect. 2.3).

Rumorama is a scalable P2P protocol that builds hierarchies of PlanetP-like [Cuenca-Acuna *et al.*, 2003] P2P networks. In Rumorama, every peer sees a portion of the network as a single, small PlanetP network and furthermore maintains connections to other peers that see other small PlanetP networks. To this end, the peer stores a small set of links pointing to neighboring peers in other subnets in order to be able to forward queries beyond the boundaries of its own PlanetP-like subnet. Each peer can choose the size of its PlanetP network according to local processing power and bandwidth capacity. Within its small PlanetP-like subnet, a peer knows resource descriptions of all other peers' data in the same subnet. These descriptions are disseminated by randomized rumor spreading and provide the basis for query routing decisions, i.e. which peers to contact in the local subnet during query processing.

Peers storing media items which are described by the criteria outlined in Fig. 1 can thus be summarized by corresponding resource or peer descriptions (cf. Fig. 2), where each peer is considered as a resource of potentially use-

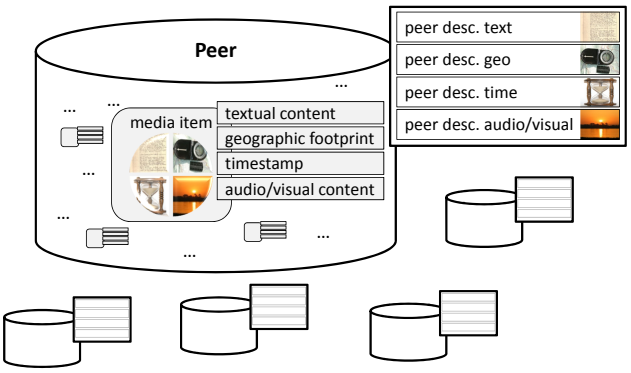


Figure 2: Peer or resource descriptions based on media item features.

ful media items with respect to the given query. These peer or resource descriptions can be envisaged as an aggregation of the features of the media items stored on the respective peer. Resource description and selection for textual data has already been extensively discussed in literature (cf. [Cuenca-Acuna *et al.*, 2003]). Techniques for time and date information are presumably less challenging and might consist of a combination of clustering (cf. [Duda *et al.*, 2000]) and histogram techniques (cf. [Ioannidis, 2003]).

We proposed resource description and selection techniques for geographic data in [Blank and Henrich, 2009; 2010]. Techniques for CBIR were e.g. addressed in [Blank *et al.*, 2007]. In this paper we will apply and adapt ultra fine-grained summaries (UFS, cf. Sect. 3), the most promising technique in the context of geographic data based on binary histograms, for CBIR and analyze its use in more detail. The contribution of this paper is *i)* the application and adaptation of UFS for CBIR, *ii)* a detailed analysis of summary sizes in the case of CBIR, and *iii)* an analysis of time complexity for peer ranking.

Of course, resource ranking based on a single criterion is only a first step. When querying for multiple criteria, e.g. for a sunset image in a certain geographic region, criterion-specific resource rankings can be combined applying a merging algorithm for ranked lists (cf. [Belkin *et al.*, 1995; Ilyas *et al.*, 2008]).

The remainder of this paper is organized as follows. Sect. 2 discusses related work. In Sect. 3 we present an analysis of the application and adaptation of UFS for CBIR. Sect. 4 concludes with an outlook on future work.

2 Related Work

2.1 Traditional Distributed IR Systems

Traditional distributed IR is mainly concerned with text data. Resources are usually described by the set of terms (or a subset of them) which are contained in the documents of a resource and some kind of frequency information per term plus possibly additional statistics [Callan, 2000]. There is plenty of work on resource selection in this context (references are e.g. given in [Thomas and Hawking, 2009; Bockting and Hiemstra, 2009]).

We will now describe approaches which address resource description and selection based on low-level visual content features. Chang et al. [Chang and Zhang, 1997; Chang *et al.*, 1997] propose three approaches. For all of them, a relatively small number of feature vectors from reference images is used (so called templates or icons). In contrast to our approach (cf. Sect. 3), templates are selected from the underlying data collection, i.e. the images that are administered by the resources. For all three approaches, a set of matching templates is computed per image and per query based on a predefined similarity threshold. The first approach uses the number of images which are assigned to a certain template as a ranking criterion. The second approach additionally applies mean and variance information w.r.t. the similarity values between a template and the images which are assigned to it. In experimental studies, both approaches perform worse than a histogram-based approach. Here, a special form of clustering is applied in order to further partition the feature space covered by a template. This approach is e.g. different to our approach w.r.t. the type of clustering, the mechanism for computing the histogram (more parameters needed, no compression, no external collection), and especially the number of applied reference points.

[Berretti *et al.*, 2004] apply a special form of hierarchical clustering to a resource's set of feature vectors in order to ascertain a resource description. A predefined maximum cluster radius is used for determining the centroids which are included in the resource description. Every path in the clustering tree is descended as long as the cluster radius of a node is bigger than the maximum cluster radius. Amongst other information, the centroids of the nodes where the search stops are included in the resource description. By varying the maximum cluster radius, the granularity and size of the resource descriptions can be adapted. When it comes to resource selection, centroids and cluster radii are applied. Compared to our approach, the size of the resource descriptions is expected to be bigger with smaller potential for compression, since centroids are usually represented by d-dimensional real-valued feature vectors (d often between 10^2 and 10^3). We proposed similar approaches in [Alali *et al.*, 2008] where local clustering and Gaussian mixture models (GMMs) are compared. For local clustering d-dimensional cluster centroids are included in the summaries. Mean and variance vectors of dimensionality d plus size information capturing the number of images of a peer which lie in a certain cluster are used as summaries in case of GMMs. GMMs perform better than local clustering in terms of ranking selectivity, but cannot outperform the approaches presented in Sect. 3. In addition, average summary sizes are expected to be bigger since the summarization of a peer with only a single real-valued image feature vector and thus one centroid will consume $d \cdot 4$ bytes for local clustering and $(2d + 1) \cdot 4$ bytes for GMMs, if we assume the usage of 4 bytes per information unit.

[Kim *et al.*, 2002] apply a multi-dimensional selectivity estimation approach based on compressed histograms (cf. [Lee *et al.*, 1999]) for resource description and selection. The d-dimensional feature space is partitioned based on a uniform, multi-dimensional grid. The histogram captures the number of features which lie in a certain bucket of the grid. Since the number of buckets rapidly increases with increasing d, multi-dimensional discrete cosine transform (DCT) is applied in order to reduce histogram sizes. With the help of an adequate sampling strategy, only the most important DCT coefficients are selected for representation. 8 bytes are used per DCT coefficient (4 bytes for the histogram index and 4 bytes for the histogram value). In their experiments in [Kim *et al.*, 2002] $2,000$ till $2,500$ coefficients are used. The entity responsible for resource selection can apply inverse DCT in order to recover the his-

togram with low error rates. For multi-dimensional range queries, the hyper-sphere representing the search region is approximated by multiple hyper-squares. Histogram information is used to determine the selectivity of individual resources w.r.t. the query. The method is further extended in order to support resource selection in heterogeneous settings, where each resource may use its own local similarity measure which may be different from a global similarity measure used by the entity performing resource selection. In [Kim and Chung, 2003], different resource selection strategies are evaluated which do not rely on histogram information and instead use queried feature vectors in order to determine relevant resources by applying different regression models on distributions of global and local similarity value pairs of queried feature vectors.

2.2 P2P IR Systems

P2P IR systems are often classified as being *structured* or *unstructured* overlay networks. As a secondary classification criterion, we introduce the distinction between *data-independent* and *data-dependent* overlays in order to reflect if a peer's content or e.g. query profiles have an effect on overlay generation. This distinction is helpful to pinpoint different characteristics in a more organized way. In the following, we will briefly discuss various approaches.

Unstructured P2P IR Systems

Data-independent: Main protocols in this group are PlanetP [Cuenca-Acuna *et al.*, 2003] and its extension Rumorama [Müller *et al.*, 2005b]. In Rumorama, a peer sees the network as a single, small PlanetP network (called subnet) with connections to other peers that see other PlanetP subnets. Each peer can choose the size of its subnet according to local processing power and bandwidth capacity. Within a subnet, a peer knows data summaries of all other peers in the same subnet. Gossiping techniques are used to disseminate the summaries. In a subnet, summary-based resource selection allows for semantic query routing. Additionally, a peer maintains a small set of links pointing to neighboring peers in other subnets in order to be able to forward queries outside the boundaries of its own subnet. In its original form, peers are assigned to subnets arbitrarily, i.e. independent of the peers' content. But, Rumorama can be easily extended by a grouping of peers similar to the content-dependent overlays described in the following. Additionally, summaries might be visualized and thus be beneficial for interactive retrieval, e.g. by providing—with low bandwidth requirements—a visual overview of peer data for a large number of peers.

Routing indexes in various forms (for references cf. [Doulkeridis *et al.*, 2009]) represent aggregated information in an unstructured network maintained at a peer for all its neighboring peers in order to decide in which direction queries should be forwarded. Initially designed for one-dimensional values in order to avoid network flooding, they have e.g. been extended to allow for multi-dimensional queries.

Data-dependent: Many semantic overlay networks (SONs) (for references and a detailed description cf. [Doulkeridis *et al.*, 2010]) can be characterized as data-dependent, unstructured P2P networks. Here, the content of a peer's data or information about past queries defines a peer's place in the network. Thus, summaries of a peer's content or query profiles are needed. Two types of links are usually maintained: short links grouping peers with similar content or query profiles into so called "clusters of interest" (COIs) and long links that are established between different COIs. During query execution the query has to be forwarded to the most promising COI(s). In order to form COIs, clustering, classification as well as gossiping techniques can be applied.

Structured P2P IR Systems

Data-independent: Structured P2P IR systems are based on distributed indexing structures with distributed hashtables (DHTs) being the most prominent class member. Every peer in the network is usually responsible for a certain range of the feature space. Thus, when entering the network or updating local content, indexing data has to be transferred to remote peers according to the peers' responsibilities. In case of data-independent, structured P2P IR systems, terms (cf. [Bender *et al.*, 2005]) or high-dimensional feature vectors for CBIR (cf. [Novak *et al.*, 2008; Lupu *et al.*, 2007; Vu *et al.*, 2009]) are usually mapped to one-dimensional or multi-dimensional keys which can be indexed in a classical DHT such as Chord [Stoica *et al.*, 2001] or CAN [Ratnasamy *et al.*, 2001] respectively.

Data-dependent: SONs—as described above—can also be implemented on top of a DHT in order to enhance query routing [Doulkeridis *et al.*, 2010]. Clustering, classification as well as gossiping techniques are applied in order to establish links to peers with similar content.

Indexing of Multiple Criteria

In structured, data-independent systems, correlations between different criteria (e.g. geographic and image content information) are difficult to exploit when indexing multiple feature types. If we e.g. assume an image from the Sahara Desert with shades of beige sand and blue sky, different peers might be responsible for indexing the geographic and the image content information. Therefore, when distributing the indexing data of the Sahara image, querying for it, or removing it from the network, (at least) two different peers have to be contacted. Within SONs, the simultaneous indexing of multiple criteria would require the definition of a similarity between peers and images combining e.g. geographic and image content information. Alternatively, multiple overlays might be maintained. Within unstructured, data-independent P2P IR systems, it is possible to apply one summary and a corresponding resource selection technique per feature type. Feature-specific peer rankings can be combined by applying an algorithm for the merging of ranked lists [Belkin *et al.*, 1995; Ilyas *et al.*, 2008]. Alternatively, the creation of summaries and resource selection algorithms integrating multiple feature types is possible (cf. [Hariharan *et al.*, 2008]).

Hybrid Approaches and Super-Peer Architectures

A main characteristic of unstructured P2P IR systems is that a peer only administers indexing data of media items which belong to its user. Thus, when entering the system or updating media items, full indexing data does not have to be transferred to remote peers. Peer autonomy is better respected compared to structured networks [Doulkeridis *et al.*, 2010]. On the other hand, structured systems offer query processing with logarithmic cost. In order to reduce the load imposed on the network when inserting new media items, super-peer architectures [Papapetrou *et al.*, 2007] as well as DHT-based indexing of compact data summaries instead of full indexing data has been proposed (cf. [Lupu *et al.*, 2007]).

In general, there is a convergence of structured and unstructured P2P IR systems with many hybrid approaches. We have e.g. evaluated an approach where indexing data is stepwisely transferred amongst peers in order to make peers more focused and—as a consequence—summaries more selective. More selective summaries with peers having specialized on a certain range of the feature space lead to more efficient resource selection [Eisenhardt *et al.*, 2008].

There is plenty of work addressing super-peer architectures (for references cf. [Doulkeridis *et al.*, 2009]). They are designed in order to overcome some limitations of "true" P2P IR systems and make use of increased capabilities such as storage capacity, processing power or available network bandwidth. Often, concepts known from "true" P2P IR systems are extended and transferred to super-peer networks. Also within super-peers the convergence of different approaches can be seen. [Doulkeridis *et al.*, 2009] e.g. apply multi-dimensional routing indexes on a super-peer level and additionally group similar super-peers close together in order to allow for better query routing.

In this context, our resource selection techniques are not restricted to data-independent, unstructured P2P IR systems. The summaries can also be used within data-dependent, unstructured P2P IR systems to form COIs and within structured networks e.g. to be indexed in a DHT. In addition, summaries could be used by super-peers for selecting either "normal" peers or other super-peers. Further application fields are also possible as will be described in the following section.

2.3 Possible Application Fields apart from the P2P Context

In addition to P2P IR (cf. Sect. 2.2) our resource summarization and selection techniques can also be used in traditional distributed IR applications (cf. Sect. 2.1). Personal meta-search is a novel application of distributed IR, where all the online resources of a person are queried (e-mail accounts, web pages, image collections, etc.). These resources are typically heterogeneous in size, media type and update frequency possibly requiring selective and space efficient summaries in this context [Thomas and Hawking, 2009].

Our summarization techniques might also be applied within (visual) sensor [Elahi *et al.*, 2009] as well as ad hoc networks [Lupu *et al.*, 2007]. Within sensor networks, limited processing power, bandwidth and energy capacities necessitate aggregation techniques which are based on local information with a clear focus on space efficiency. [Lupu *et al.*, 2007] present an approach for ad hoc information sharing based on mobile devices when people meet at certain events or places. Here, it might not be feasible to transfer complete indexing data but only summarized information.

Distributed IR techniques can also be used for vertical selection within aggregated search [Arguello *et al.*, 2009]. Vertical selection is the task of identifying relevant verticals, i.e. focused search services such as image, news, video or shopping search. A user issuing a textual query "music beatles" might also be interested in music videos and thus the results of video search or small previews should be integrated in result presentation of classical web search. In this context, a vertical can be interpreted as a resource and the task of selecting relevant verticals is similar to resource selection in distributed IR requiring adequate features, i.e. resource descriptions, and corresponding selection mechanisms.

Space efficient resource descriptions might also be beneficial in the context of recommender systems and social search e.g. in order to compute the similarity between different users of social network sites. Similar users can be determined not only based on having the same friends, using the same tags, bookmarking the same media items, etc. [Guy *et al.*, 2010], but also depending on the similarity of media content.

Another potential application area is automatic theme identification. Automatic theme identification of photo sets e.g. in case of digital print products[1] is concerned with the task of finding suitable background themes for a given set of images. Themes can be travel, wedding, etc. Each theme can be described by a set of photos and modeled as a summary. Afterwards, the theme descriptions and the description of a user's collection can be compared in order to recommend the best matching theme(s).

Resource selection techniques have been successfully applied to blog site search [Elsas *et al.*, 2008]. A blog feed is viewed as a single collection and individual posts are interpreted as documents in order to retrieve similar blogs according to a given information need. A similar approach can be undertaken in passage retrieval such as XML retrieval where different sections, subsections, etc. might be grouped together as a resource (for references cf. [Lalmas, 2009]).

Also expert search [Balog *et al.*, 2009] could presumably be built based on resource description and selection techniques. Here, a user is interested in finding human experts in an enterprise for example. Thus, e.g. all documents a person has (co)authored could be modeled as a resource and finding an expert would result in selecting the most promising resource.

Compact resource descriptions might also be valuable for focused crawling [Ahlers and Boll, 2009]. If a service provides summaries of the image content of a certain website or media archive, a crawler could estimate the potential usefulness of this resource for its focused crawling task before actually visiting the source. This way, crawl efficiency can be improved by preventing the crawler from analyzing too many irrelevant pages. Web traffic imposed by downloading large sets of images in order to extract CBIR features can thus be avoided.

Tree-based index structures are also related to our work (cf. [Samet, 2006]). The decision of choosing the best subtree is similar to the resource selection problem. Summaries in the P2P context correspond to aggregations maintained in the nodes of a tree, e.g. bounding boxes in the case of an R-tree [Guttman, 1984].

3 Resource Summarization and Selection for low-level Visual Content Features

[Müller *et al.*, 2005a] proposed the use of "cluster histograms" for distributed CBIR. In order to compute cluster histograms, a moderate number of reference points is used (e.g. $k = 256$). This set of reference points is known to all peers. Every image feature vector of a peer's local image collection is assigned to the closest reference point. Hereby, a cluster histogram is computed counting how many image feature vectors of a peer's collection are

[1] *http://comminfo.rutgers.edu/conferences/mmchallenge/2010/03/11/cewe-challenge/*, last visit: 8.7.2010

closest to a certain reference point, i.e. cluster centroid c_j ($1 \leq j \leq k$). Reference points are determined by distributed k-Means clustering which imposes some load on the network. During resource selection only histogram information regarding the cluster whose reference point lies closest to the query feature vector is used. Peers with more feature vectors assigned to this cluster are ranked higher than peers with fewer feature vectors assigned to the cluster.

In [Eisenhardt *et al.*, 2006] the performance of resource selection is further improved. A list *L* of reference points c_j is sorted in ascending order according to the distance of c_j to the query feature vector q. In order to rank peer p_a before p_b or vice versa *L* is processed from the beginning possibly till the end. The first element of *L* corresponds to the cluster centroid being closest to q. A peer with more documents in this cluster is ranked higher than a peer with fewer documents in the very cluster. If two peers p_a and p_b administer the same amount of images in the analyzed cluster and the end of the list has not yet been reached, the next element out of *L* is chosen and based on the number of documents within the current cluster it is again tried to rank p_a before p_b or vice versa. As a second modification, distributed clustering is replaced in [Eisenhardt *et al.*, 2006] by a random selection of reference points. Overall ranking selectivity is slightly affected, but there is no longer any network load imposed due to distributed clustering.

We have extended the work from [Müller *et al.*, 2005a; Eisenhardt *et al.*, 2006] in several directions (cf. [Blank *et al.*, 2007]). First, within highly fine-grained summaries (HFS_k with k indicating the number of reference points used) we increased the number of reference points for computing the cluster histogram, e.g. from $k = 256$ to $k = 8,192$ or even more. By doing so, the feature space is partitioned in a more fine-grained way offering improved ranking selectivity. Since a higher number of reference points would lead to less space efficient summaries, we apply compression techniques. Thus, we can achieve better ranking selectivity with more space efficient resource descriptions compared to the approaches in [Müller *et al.*, 2005a; Eisenhardt *et al.*, 2006]. The average size of a peer's summary information is approx. 110 bytes for $k = 16,384$, which is clearly less compared to other approaches in distributed CBIR (cf. Sect. 2.1) if they were applied to our scenario directly. Second, within our approach reference points are selected from an external source and transferred to peers together with updates of the P2P software. This leads to a decrease in overall network load and makes distributed selection mechanisms obsolete. Ranking selectivity is slightly affected by this change as will be shown in Sect. 3.3.

3.1 Experimental Setup

In the experiments we use a 166-dimensional uniformly quantized color histogram[2] based on the HSV color space with 18 hues, 3 saturations and 3 values, plus 4 levels of gray. Image feature vectors are compared using Euclidean distance. We perform 20 runs where we change the reference points used. During a run we perform 100 queries where we randomly select a query image from the underlying collection. The set of queries stays constant over all runs. By analyzing the number of peers which are contacted on average in order to retrieve the 20 closest feature vectors w.r.t. a given query feature vector we assume that the most similar images are the ones the user is interested in. We crawled a collection of $233,827$ Flickr images. They are assigned to peers based on the Flickr user ID in order to reflect a realistic scenario. Hence, we assume that every Flickr user operates a peer of its own. The images are mapped to $10,601$ peers/users which are used in our simulation. Fig. 3 shows the distribution of peer sizes, i.e. the number of images which are maintained per peer. The general characteristic is typical for P2P file sharing applications with few peers managing large amounts of the images and many peers administering only few images [Saroiu *et al.*, 2002].

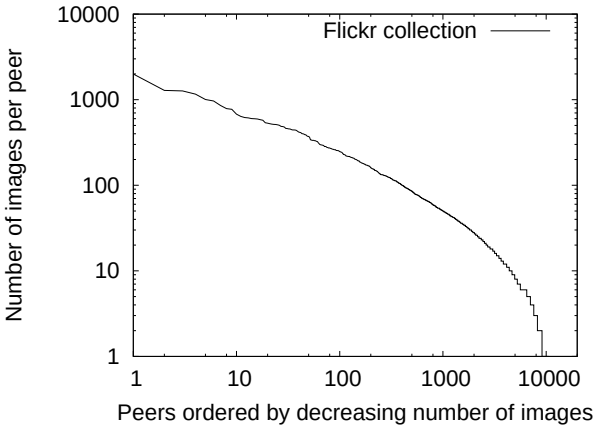


Figure 3: Distribution of peer sizes.

3.2 Using UFS for CBIR

When summarizing geographic footprints, binary histograms (so called UFS_k: ultra fine-grained summaries) outperformed HFS_k (cf. [Blank and Henrich, 2010]). In contrast to HFS, UFS are based on a bit vector with the bit at position j indicating if centroid j is the closest centroid to one or more of a peer's image feature vectors. Hence, we obtain a bit vector of size k. Of course, there is some loss of information when switching from HFS to UFS with k staying constant. However, UFS have the potential of resulting in more space efficient resource descriptions. Potentially, this allows for more centroids being used which might result in similar or even improved ranking selectivity compared to HFS. In the following we will thus evaluate the use of UFS in the context of high-dimensional feature vectors for CBIR.

3.3 Analysis of Ranking Selectivity

Reference points for summary creation and peer ranking are chosen from the underlying collection (UFS/HFS) or a second collection of $45,931$ Flickr images (UFSe/HFSe with "e" indicating the use of an external collection for the reference points). It is important to note that both collections are disjoint w.r.t. the unique Flickr image and user IDs, but there is some minor natural overlap amongst collections w.r.t. image content; 24 of the $233,827$ images also appear in the external collection, because some images are uploaded by multiple users independently on Flickr.

Fig. 4 shows the number of peers which are contacted on average in order to retrieve the 20 closest feature vectors w.r.t. a given query feature vector. Ranking selectivity increases degressively with increasing k. There is a gap in ranking selectivity when choosing the reference points

[2] *http://www.gnu.org/software/gift/*, last visit: 5.7.2010

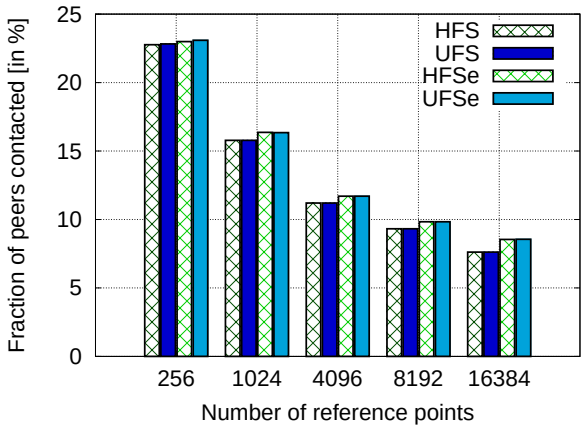


Figure 4: Fraction of peers contacted to retrieve top-20 image feature vectors (ranking selectivity).

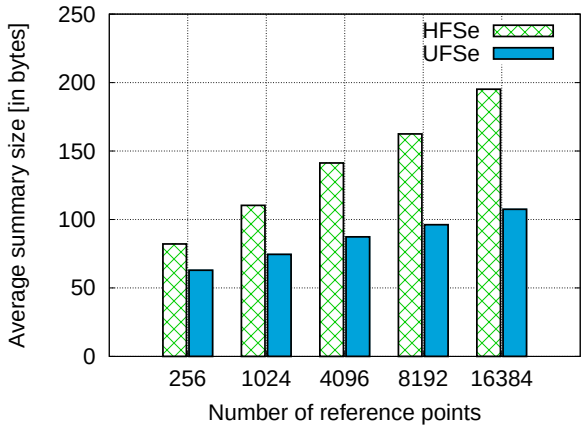


Figure 5: Avg. summary sizes (zipped).

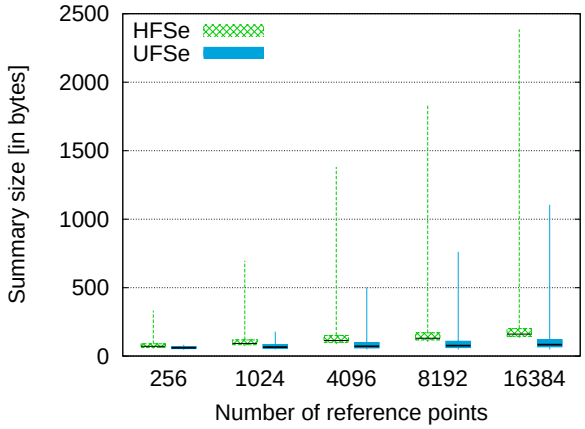


Figure 6: Box plot of summary sizes.

from an external collection (HFSe compared to HFS and UFSe compared to UFS). The gap increases with increasing k. For UFS/HFS, with increasing k also the probability of choosing a centroid which is used also as a query feature vector increases. Such situations might lead to improved ranking selectivity, since queries are randomly chosen from the underlying data collection. The evaluation of other sources of feature vectors as queries will be part of future work. Fig. 4 additionally shows slightly improved ranking selectivity for HFS(e)$_{256}$ compared to UFS(e)$_{256}$ respectively[3], which is due to the use of non-binary histogram information during peer ranking within HFS(e). In general, this gap more and more diminishes when increasing the number of centroids used since HFS(e) histograms more and more pass into binary histograms. Already for HFS$_{1,024}$ compared to UFS$_{1,024}$ and HFSe$_{1,024}$ compared to UFSe$_{1,024}$ there is no noticeable difference in ranking selectivity at all.

3.4 Analysis of Summary Sizes

The size of the resource descriptions after zipping is analyzed in Fig. 5 and Fig. 6. Fig. 5 shows average summary sizes s_{avg} when using UFSe instead of HFSe. The plot for UFS and HFS shows similar characteristics. In addition, Fig. 6 visualizes the different quartiles and minimum/maximum values of the summary sizes in a box plot. It shows that the median in case of HFSe is bigger compared to UFSe. Interquartile ranges of HFSe and UFSe become more and more similar when increasing k although the overall range of HFSe summary sizes is greater than the range of summary sizes in case of UFSe. All distributions of summary sizes are positively skew indicating many peers with small summary sizes and few peers with big summary sizes. Thus, the distribution of peer sizes (cf. Fig. 3) is reflected in the distribution of summary sizes (cf. Fig. 6).

One might think of a hybrid peer ranking scheme e.g. using HFSe for the smaller peers (i.e. peers with few documents) and UFSe for the bigger peers (i.e. peers with many documents) in order to reduce network load imposed by rumor spreading. The cost of one round of rumor spreading can be estimated by $s_{avg} \cdot n \cdot (n-1)$ with n being the number of peers in a PlanetP-like network. Hence, the cost is proportional to s_{avg}. Since an increase in ranking selectivity can be perceived only for small values of k when switching from HFS(e) to UFS(e) respectively (cf. Fig. 4), the UFS(e) alternative can be safely chosen for all peers in the network in case of big values of k. If there are bigger differences in ranking selectivity amongst competing resource description and peer ranking schemes, the cost for query processing has to be additionally taken into account. A more detailed analysis can be found in [Blank and Henrich, 2010].

3.5 Analysis of Time Complexity

In general, it is important that peer ranking can be done within a reasonable amount of time. As described above, ranking peers mainly means sorting k-dimensional numbers where the importance of the single dimensions is defined by the list L which contains the reference points sorted according to their distance to the query feature vector. In a first run the peers are sorted w.r.t. the dimension representing the closest reference point. Of course, this sorting can be done in $\mathcal{O}(n \cdot \log n)$ where n stands for the number of peers in the considered PlanetP network. In a worst case scenario all peers would be identical in the number of media items maintained in each of the k clusters ending up in a complexity of $\mathcal{O}(k \cdot n \cdot \log n)$. Thus, the worst case complexity for calculating a peer ranking depends on k which of course is disadvantageous for HFS(e)/UFS(e) with high values of k.

[3] HFS(e)$_k$ is used as an abbreviation for "HFS$_k$ and HFSe$_k$". The same notation is also adopted for UFS throughout the paper. In a similar way, HFS/UFS also abbreviates "HFS and UFS".

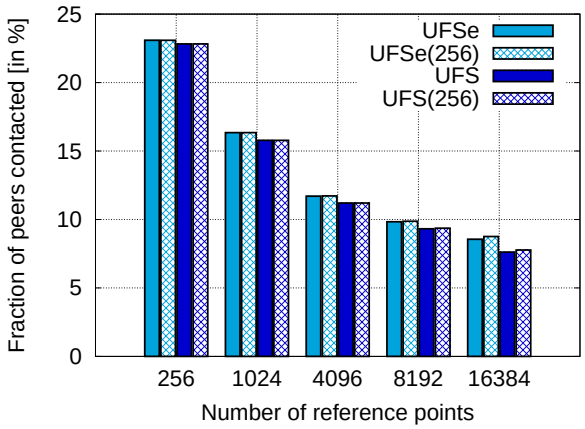


Figure 7: Fraction of peers contacted for retrieving top-20 images with UFS(e)(256) vs. UFS(e) (ranking selectivity).

In order to test whether this worst case scenario has practical implications, we compared the original approach considering the clusters to the end, if necessary, with a modified variant using at most the 256 clusters closest to the query. If no decision is possible after considering the histogram values for these 256 clusters, a random choice is made. In the following UFS(e)$_k$(256) will denote the modified approach considering k centroids for summary creation while at most only applying the 256 closest centroids w.r.t. the query feature vector for peer ranking. Results can be seen in Fig. 7. There is no noticeable difference in ranking selectivity for UFS(e)$_k$ compared to UFS(e)$_k$(256) respectively for summaries with up to $k = 8,192$ centroids. UFS(e)$_{16,384}$ performs slightly better than UFS(e)$_{16,384}$(256). When increasing k, the feature space is partitioned in a more fine grained way. If only 256 centroids are used for peer ranking, the fraction of unused centroids which potentially contain relevant information increases, e.g. in case of UFS(e)$_{16,384}$(256), $1 - \frac{256}{16,384} = 98.4\%$ of summary information is discarded during peer ranking.

The results in Fig. 7 demonstrate two things. First, obviously very few of the k histogram bins are usually considered for peer ranking. Otherwise the differences between UFS(e) and UFS(e)(256) would have been higher. Second, programmers anxious about worst case bounds can stop processing after considering a certain number of histogram bins and thus avoid the worst case of $\mathcal{O}(k \cdot n \cdot \log n)$.

4 Conclusion & Outlook

In this paper we have applied and adapted binary histograms which were originally designed for the summarization of low-dimensional spatial data for resource description and selection based on high-dimensional CBIR features. Compared to earlier work, summaries can be zipped more efficiently. A huge number of reference points (e.g. $16,384$) is applied in order to generate resource descriptions. We have shown that it is possible to use only a small fraction of reference points (e.g. 256) during peer ranking in order to speed-up query processing with only a marginal decrease in ranking selectivity.

In future work we will try to find an adequate stopping criterion which indicates when it is no longer beneficial to contact further peers. This might be woven with a technique that adaptively determines the number of centroids used for peer ranking. Additionally we will apply our resource descriptions in order to summarize local image features such as SIFT.

References

[Ahlers and Boll, 2009] Dirk Ahlers and Susanne Boll. Adaptive geospatially focused crawling. In *Proc. of the 18th ACM Conf. on Information and Knowledge Management*, pages 445–454, Hong Kong, China, 2009.

[Allali *et al.*, 2008] Soufyane Allali, Daniel Blank, Wolfgang Müller, and Andreas Henrich. Image data source selection using Gaussian mixture models. In *Adaptive Multimedial Retrieval: Retrieval, User, and Semantics: 5th Intl. Workshop, AMR 2007, Paris, France, 2007, Revised Selected Papers*, pages 170–181, Berlin, Heidelberg, 2008. Springer LNCS 4918.

[Arguello *et al.*, 2009] Jaime Arguello, Fernando Diaz, Jamie Callan, and Jean-Francois Crespo. Sources of evidence for vertical selection. In *Proc. of the 32nd Intl. ACM SIGIR Conf. on research and development in Information Retrieval*, pages 315–322, Boston, MA, USA, 2009.

[Balog *et al.*, 2009] Krisztian Balog, Ian Soboroff, Paul Thomas, Nick Craswell, Arjen de Vries, and Peter Bailey. Overview of the TREC 2008 enterprise track. In *The Seventeenth Text Retrieval Conf. Proceedings (TREC 2008)*. NIST, 2009. Special Publication.

[Belkin *et al.*, 1995] N. J. Belkin, P. Kantor, E. A. Fox, and J. A. Shaw. Combining the evidence of multiple query representations for information retrieval. *Inf. Processing and Management*, 31(3):431–448, 1995.

[Bender *et al.*, 2005] Matthias Bender, Sebastian Michel, Gerhard Weikum, and Christian Zimmer. The minerva project: Database selection in the context of P2P search. In *Datenbanksysteme in Business, Technologie und Web, 11. Fachtagung des GI-Fachbereichs "Datenbanken und Informationssysteme" (DBIS)*, pages 125–144, Karlsruhe, Germany, 2005.

[Berretti *et al.*, 2004] Stefano Berretti, Alberto Del Bimbo, and Pietro Pala. Merging results for distributed content based image retrieval. *Multimedia Tools Appl.*, 24(3):215–232, 2004.

[Blank and Henrich, 2009] Daniel Blank and A. Henrich. Summarizing georeferenced photo collections for image retrieval in P2P networks. In *Proc. of Workshop on Geographic Information on the Internet*, pages 55–60, `http://georama-project.labs.exalead.com/workshop/GIIW-proceedings.pdf` (12.11.2009), Toulouse, France, 2009.

[Blank and Henrich, 2010] Daniel Blank and Andreas Henrich. Description and selection of media archives for geographic nearest neighbor queries in P2P networks. In *Proc. of IAPMA2010: Information Access for Personal Media Archives Workshop*, pages 22–29, `http://doras.dcu.ie/15373/` (25.5.2010), Milton Keynes, UK, 2010.

[Blank *et al.*, 2007] Daniel Blank, Soufyane El Allali, Wolfgang Müller, and Andreas Henrich. Sample-based creation of peer summaries for efficient similarity search in scalable peer-to-peer networks. *ACM SIGMM Workshop on Multimedia Information Retrieval (MIR 2007), Augsburg, Germany*, pages 143–152, 2007.

[Bockting and Hiemstra, 2009] Sander Bockting and Djoerd Hiemstra. Collection Selection with Highly Discriminative Keys. In *Proc. of the 7th Workshop on Large-Scale Distributed Systems for Information Retrieval*, `http://lsdsir09.isti.cnr.it/lsdsir09-1.pdf` (26.04.2010), Boston, MA, USA, 2009.

[Callan, 2000] Jamie Callan. Distributed information retrieval. In W. B. Croft, editor, *Advances in Information Retrieval*, pages 127–150. Kluwer Academic Publishers, 2000.

[Chang and Zhang, 1997] Wendy Chang and Aidong Zhang. Metadata for distributed visual database access. In *2nd IEEE Metadata Conf.*, pages 1–11, Silver Spring, MD, USA, 1997.

[Chang *et al.*, 1997] Wendy Chang, Gholamhosein Sheikholeslami, Aidong Zhang, and Tanveer F. Syeda-Mahmood. Efficient resource selection in distributed visual information systems. In *Proc. of the 5th ACM Intl. Conf. on Multimedia*, pages 203–213, Seattle, Washington, USA, 1997.

[Cuenca-Acuna *et al.*, 2003] Francisco Cuenca-Acuna, Christopher Peery, Richard P. Martin, and Thu D. Nguyen. PlanetP: Using gossiping to build content addressable peer-to-peer information sharing communities. In *IEEE Intl. Symp. on High Performance Distributed Computing*, pages 236–246, Seattle, WA, USA, 2003.

[Doulkeridis *et al.*, 2009] Christos Doulkeridis, Akrivi Vlachou, Kjetil Nørvåg, Yannis Kotidis, and Michalis Vazirgiannis. Multidimensional routing indices for efficient distributed query processing. In *Proc. of the 18th ACM Conf. on Information and Knowledge Management*, pages 1489–1492, Hong Kong, China, 2009.

[Doulkeridis *et al.*, 2010] Christos Doulkeridis, Akrivi Vlachou, Kjetil Nørvåg, and Michalis Vazirgiannis. *Handbook of Peer-to-Peer Networking*. Part 4: Distributed Semantic Overlay Networks. Springer Science+Business Media, 1st edition, 2010.

[Duda *et al.*, 2000] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification*. Wiley-Interscience, 2nd edition, Nov. 2000.

[Eisenhardt *et al.*, 2006] Martin Eisenhardt, Wolfgang Müller, Andreas Henrich, Daniel Blank, and Soufyane El Allali. Clustering-based source selection for efficient image retrieval in peer-to-peer networks. In *8th IEEE Intl. Symp. on Multimedia*, pages 823–830, San Diego, CA, USA, 2006.

[Eisenhardt *et al.*, 2008] Martin Eisenhardt, Wolfgang Müller, Daniel Blank, Soufyane El Allali, and Andreas Henrich. Clustering-based, load balanced source selection for CBIR in P2P networks. *Intl. Journal of Semantic Computing (IJSC)*, 2(2):235–252, 2008.

[Elahi *et al.*, 2009] B. Maryam. Elahi, Kay Römer, Benedikt Ostermaier, Michael Fahrmair, and Wolfgang Kellerer. Sensor ranking: A primitive for efficient content-based sensor search. In *Intl. Conf. on Information Processing in Sensor Networks*, pages 217–228, Washington, DC, USA, 2009. IEEE.

[Elsas *et al.*, 2008] Jonathan L. Elsas, Jaime Arguello, Jamie Callan, and Jaime G. Carbonell. Retrieval and feedback models for blog feed search. In *Proc. of 31st Intl. ACM SIGIR Conf. on research and development in Information Retrieval*, pages 347–354, Singapore, 2008.

[Guttman, 1984] Antonin Guttman. R-trees: A dynamic index structure for spatial searching. In *ACM SIGMOD Conf.*, pages 47–57, Boston, MA, 1984. ACM.

[Guy *et al.*, 2010] Ido Guy, Michal Jacovi, Adam Perer, Inbal Ronen, and Erel Uziel. Same places, same things, same people?: mining user similarity on social media. In *CSCW '10: Proc. of ACM Conf. on Computer Supported Cooperative Work*, pages 41–50, Savannah, Georgia, USA, 2010.

[Hariharan *et al.*, 2008] Ramaswamy Hariharan, Bijit Hore, and Sharad Mehrotra. Discovering GIS sources on the web using summaries. In *Proc. of 8th ACM/IEEE joint Conf. on digital libraries*, pages 94–103, Pittsburgh, PA, USA, 2008. ACM.

[Ilyas *et al.*, 2008] Ihab F. Ilyas, George Beskales, and Mohamed A. Soliman. A survey of top-k query processing techniques in relational database systems. *ACM Comput. Surv.*, 40(4):1–58, 2008.

[Ioannidis, 2003] Yannis Ioannidis. The history of histograms (abridged). In *Proc. of 29th Intl. Conf. on Very Large Data Bases*, pages 19–30, Berlin, Germany, 2003.

[Kim and Chung, 2003] Deok-Hwan Kim and Chin-Wan Chung. Collection fusion using Bayesian estimation of a linear regression model in image databases on the web. *Inf. Processing and Management*, 39:267–285, 2003.

[Kim *et al.*, 2002] Deok-Hwan Kim, Seok-Lyong Lee, and Chin-Wan Chung. Heterogeneous image database selection on the web. *The Journal of Systems and Software*, 64:131–149, 2002.

[Lalmas, 2009] Mounia Lalmas. *XML retrieval*. Synthesis Lectures on Information Concepts, Retrieval and Services. Morgan & Claypool Publishers, 2009.

[Lee *et al.*, 1999] Ju-Hong Lee, Deok-Hwan Kim, and Chin-Wan Chung. Multi-dimensional selectivity estimation using compressed histogram information. *SIGMOD Record*, 28(2):205–214, 1999.

[Lupu *et al.*, 2007] Mihai Lupu, Jianzhong Li, Beng Chin Ooi, and Shengfei Shi. Clustering wavelets to speed-up data dissemination in structured P2P manets. In *Intl. Conf. on Data Engineering*, pages 386–395, Istanbul, Turkey, 2007. IEEE.

[Müller *et al.*, 2005a] W. Müller, M. Eisenhardt, and A. Henrich. Fast retrieval of high-dimensional feature vectors in P2P networks using compact peer data summaries. *Multimedia Systems*, 10(6):464–474, 2005.

[Müller *et al.*, 2005b] Wolfgang Müller, Martin Eisenhardt, and Andreas Henrich. Scalable summary based retrieval in P2P networks. In *Proc. of the 14th ACM Conf. on Information and Knowledge Management*, pages 586–593, Bremen, Germany, 2005.

[Novak *et al.*, 2008] David Novak, Michal Batko, and Pavel Zezula. Web-scale system for image similarity search: When the dreams are coming true. In *Intl. Workshop on Content-Based Multimedia Indexing*, pages 446–453, London, UK, 2008. IEEE.

[Papapetrou *et al.*, 2007] Odysseas Papapetrou, Wolf Siberski, Wolf-Tilo Balke, and Wolfgang Nejdl. DHTs over peer clusters for distributed information retrieval. In *21st Intl. Conf. on Advanced Information Networking and Applications*, pages 84–93, Niagara Falls, Canada, 2007. IEEE.

[Ratnasamy *et al.*, 2001] Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard Karp, and Scott Schenker. A scalable content-addressable network. In *ACM SIGCOMM*, pages 161–172, San Diego, CA, USA, 2001.

[Samet, 2006] H. Samet. *Foundations of Multidimensional and Metric Data Structures*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2006.

[Saroiu *et al.*, 2002] Stefan Saroiu, P. Krishna Gummadi, and Steven D. Gribble. A measurement study of peer-to-peer file sharing systems. In *ACM/SPIE Multimedia Computing and Networking*, pages 156–170, San Jose, CA, USA, 2002.

[Stoica *et al.*, 2001] Ion Stoica, Robert Morris, David Karger, Frans Kaashoek, and Hari Balakrishnan. Chord: A scalable peer-to-peer lookup service for Internet applications. In *SIGCOMM'01*, pages 149–160, San Diego, CA, USA, 2001.

[Thomas and Hawking, 2009] Paul Thomas and D. Hawking. Server selection methods in personal metasearch: a comparative empirical study. *Information Retrieval*, 12(5):581–604, 2009.

[Vu *et al.*, 2009] Quang Hieu Vu, Mihai Lupu, and Sai Wu. Simpson: Efficient similarity search in metric spaces over P2P structured overlay networks. In *Proc. of 15th Intl. Euro-Par Conf. on Parallel Processing*, pages 498–510, Delft, The Netherlands, 2009. Springer.

Benutzerorientiertes Dokumenten-Clustering durch die Verwendung einer Anfragemenge

Marc Lechtenfeld
Informationssysteme
Universität Duisburg-Essen

Abstract

Der Einsatz einer Anfragemenge ermöglicht die Berücksichtigung des Informationsbedürfnisses des Benutzers beim Strukturierungsprozess von Dokumenten. Der vorgestellte benutzerorientierte Ansatz kann den Benutzer damit bei mehreren Suchaktivitäten unterstützen. Geplante Benutzerexperimente sollen die Effektivität, Effizienz und Benutzerzufriedenheit prüfen.

1 Einführung

Ein Benutzer kann mithilfe zahlreicher Information-Retrieval-Verfahren nach wohlspezifizierten Informationen suchen, indem er sein Informationsbedürfnis in Form einer Anfrage an sie richtet. Er ist jedoch nicht immer in der Lage sein Informationsbedürfnis genau zu spezifizieren und in einer speziellen Systemanfrage auszudrücken. Um seine Aufgabe in diesem Fall trotzdem erfüllen zu können, ist er dann auf die Unterstützung weiterer Suchaktivitäten durch das System angewiesen [Belkin, 1993].

Explorative Suchaktivitäten

Das Durchstöbern einer von einem Clustering-Verfahren strukturierten Dokumentmenge stellt für den Benutzer eine Möglichkeit dar, nach relevanten Dokumenten zu suchen, ohne dabei sein Informationsbedürfnis explizit ausdrücken zu müssen. Diese Dokumentmenge könnte beispielsweise eine große Kollektion von Dokumenten sein, die vom Benutzer über die in ihr enthaltenen Themen durchsucht werden kann oder ein Suchergebnis zu einer mehrdeutigen Anfrage, das durch die Gruppierung der Dokumente mit gleicher Bedeutung für den Benutzer verständlicher wird.

Benutzerorientierung

Klassische Dokumenten-Clustering-Verfahren strukturieren Dokumentmengen allerdings häufig nur nach thematischen Gesichtspunkten. Das konkrete Informationsbedürfnis des Benutzers wird nicht direkt in den Strukturierungsprozess einbezogen. Ein Benutzer kann sich jedoch für unterschiedliche Aspekte eines Dokuments interessieren, beispielsweise für die Textsorte, die Verständlichkeit oder die inhaltliche Qualität eines Dokuments. So möchte er für eine Sammlung von Zeitungsartikeln vielleicht herausfinden, über welche Ereignisse in den unterschiedlichen Ressorts negativ berichtet wurde. Dies erfordert beispielsweise sowohl eine thematische Gruppierung der Dokumente nach Ressorts (Politik, Sport, usw.) als auch eine Gruppierung nach Sentiment [Pang *et al.*, 2002] (positive, neutrale und negative Äußerungen).

2 Benutzerorientiertes Dokumenten-Clustering

Um ein für den Benutzer nützliches Clustering zu erzeugen, sollte sein Informationsbedürfnis den Strukturierungsprozess auf eine grundlegende Art und Weise beeinflussen.

2.1 Optimum Clustering Framework

Die theoretische Basis dazu bildet das *Optimum Clustering Framework* (OCF) [Fuhr *et al.*, 2010]. Zur Bestimmung der Dokumentähnlichkeit führt es eine Anfragemenge ein, die in Verbindung mit einem Retrievalmodell dazu verwendet wird, die Relevanz eines Dokuments für den Benutzer zu schätzen. Die Ähnlichkeit zweier Dokumente ergibt sich dann aus dem Skalarprodukt der zwei Dokumentrepräsentationen. Diese beiden Vektoren bestehen aus den Relevanzwahrscheinlichkeiten bezüglich aller Anfragen der Anfragemenge. Dokumente werden demnach als ähnlich definiert, wenn sie bezüglich möglichst vieler Anfragen gleichzeitig potenziell relevant sind.

Klassische Dokumenten-Clustering-Verfahren verwenden dabei eine Anfragemenge, die aus allen in der Dokumentmenge vorkommenden Termen besteht. Experimentelle Ergebnisse zeigen jedoch, das Clusterings besser werden, wenn die verwendeten Anfragen stärker den tatsächlichen Anfragen entsprechen, die der Benutzer für eine anfrageorientierte Suche verwenden würde.

2.2 Berücksichtigung einer Anfragemenge

Um eine Strukturierung der Dokumentmenge zu erzeugen, die sich stärker am Informationsbedürfnis des Benutzers orientiert und damit nützlicher für ihn ist, kann daher das Optimum Clustering Framework auf eine Anfragemenge angewandt werden, die an das Informationsbedürfnis des Benutzers stärker angepasst ist.

Die Anfragemenge sollte dabei so gestaltet sein, dass sie die Anfragen, die das Informationsbedürfnis des Benutzers am besten ausdrücken, enthält, aber nicht wesentlich größer ist. Beim Clustering eines Suchergebnisses kann diese Anfragemenge beispielsweise auch alle potenziellen Anfrageerweiterungen der Suchanfrage enthalten.

2.3 Bestimmung der Anfragemenge

Liegen über das Informationsbedürfnis des Benutzers keinerlei Informationen vor, so erscheint es sinnvoll als Anfragemenge die Menge aller in der Dokumentmenge vorkommenden Terme als Eintermanfragen zu verwenden. Dies entspricht dann dem klassischen Clustering nach dem dominantesten Aspekt, der in der Regel das Thema ist. Da eine spezialisierte Anfragemenge jedoch zu besseren Strukturierungen führt, sollte, wenn weitere Hinweise

über das konkrete Informationsbedürfnis des Benutzers verfügbar sind, eine Anfragemenge verwendet werden, die an diese Informationen angepasst ist.

Anfragemengen für bestimmte Aspekte
Durch die Interaktion mit dem Benutzer können besser an sein Informationsbedürfnis angepasste Anfragemengen gefunden werden. Eine Möglichkeit dazu besteht darin, den Benutzer explizit nach den Aspekten zu fragen, für die er sich interessiert. Beispielsweise durch Vorlage einer Liste von Aspekten, die allgemein für viele Benutzer interessant sind und die in der vorliegenden Dokumentmenge vorkommen. So könnte eine Kollektion von Büchern z. B. nach den Themengebieten, nach den Leserbewertungen oder nach der Verständlichkeit gruppiert werden. Zur Strukturierung wird dann jeweils die vorbereitete Anfragemenge eingesetzt, die den Aspekt am besten beschreibt, den der Benutzer ausgewählt hat. Durch die Möglichkeit zur Auswahl verschiedener Aspekte wird so ein mehrdimensionales Clustering aufgrund unterschiedlicher Aspekte möglich.

Individuelle Anfragemengen
Das System könnte während der Interaktion mit dem Benutzer jedoch auch Informationen über den Benutzer indirekt sammeln. Diese könnten dabei helfen die anfänglich große Anfragemenge, die ein allgemeines Informationsbedürfnis ausdrückt, sukzessive zu spezialisieren und so an das individuelle Informationsbedürfnis des Benutzers anzupassen. Auf diese Weise wäre es vielleicht möglich, automatisch zu bestimmen, für welche Aspekte sich der Benutzer wahrscheinlich interessiert.

Mögliche Quellen für die Eingrenzung der potenziellen Anfragen sind neben dem Interaktionsverhalten des Benutzers mit dem System beispielsweise auch Charakteristika der Kollektion, Adaptierungen der Benutzerschnittstelle, die verwendeten Retrievalmodelle oder externe Quellen wie z. B. die Enzyklopädie *Wikipedia*.

2.4 Unterstützte Suchaktivitäten

Die Kombination von Verfahren des Information-Retrieval und Dokumenten-Clustering ermöglicht die Unterstützung verschiedener Suchaktivitäten.

Interpretation einer Dokumentmenge
Einen ersten Überblick über eine unbekannte Kollektion oder über ein Suchergebnis kann sich der Benutzer mithilfe einer Strukturierung dieser Dokumentmengen verschaffen. So könnte man durch eine Gruppierung der Ergebnisse einer Buchsuche beispielsweise einen Überblick über aktuelle Programmiersprachen erhalten.

Die Möglichkeit die Strukturierung aufgrund unterschiedlicher Aspekte durchführen zu lassen, kann dem Benutzer dabei helfen die Beschaffenheit der Dokumentmenge oder sein Informationsbedürfnis besser zu verstehen. Die Bücher über Programmiersprachen lassen sich beispielsweise auch nach den Vorkenntnissen z. B. in Bücher für Anfänger und Fortgeschrittene oder nach der Leserzufriedenheit über die Rezensionen gruppieren.

Über die jeder Gruppierung zugrundeliegende Anfragemenge ist für den Benutzer möglicherweise erkennbar, für welche Anfragen oder Aspekte die Dokumente der gerade betrachteten Gruppierung relevante Informationen liefern und wie sich die Dokumente der unterschiedlichen Cluster voneinander unterscheiden. Auf diese Weise kann er lernen, über welche Anfragen die gewünschten Dokumente vom Retrievalsystem zurückgeliefert werden bzw. wie er eine Anfrage mit weiteren Termen ergänzen könnte, um die gewünschten Dokumente zu erhalten. Beim Clustering eines Suchergebnisses kann man die Fragemenge beispielsweise als Menge von möglichen Frageerweiterungen betrachten.

Finden durch Erkennen
Durch das Durchstöbern der gruppierten Dokumentmenge kann der Benutzer Dokumente finden, ohne sein Informationsbedürfnis explizit ausdrücken zu müssen. Er kann relevante Dokumente entdecken, indem er die Dokumente – nach der Terminologie von Belkin [Cool and Belkin, 2002] – *scannt* und die relevanten Dokumente in der Dokumentmenge als für ihn relevant *erkennt*. So könnte der Benutzer bei einer Suche nach einem Einsteigerbuch für eine neue Programmiersprache z. B. auf eine Untergruppe von Umsteigerbüchern stoßen, die für ihn nützlicher sind.

Durch die Interaktion mit dem System etwa durch die Auswahl einzelner Cluster oder durch das manuelle Gruppieren einzelner Dokumente, kann das System die Fragemenge automatisch Schritt für Schritt an das Informationsbedürfnis des Benutzers anpassen. Auf diese Weise kann ein auf den Benutzer zugeschnittenes Clustering der Kollektion oder der Ergebnismenge erzeugt werden.

3 Benutzerorientierte Evaluation

Es sind Benutzerexperimente geplant, die die Nützlichkeit dieses Ansatzes für den Benutzer bestimmen sollen.

Es sollen dabei insbesondere auch Informationsbedürfnisse untersucht werden, die sich nur schwer in Form einer spezifischen Anfrage ausdrücken lassen. Dabei ist zu berücksichtigen, dass sich der Benutzer für verschiedene Aspekte interessieren kann, die zu unterschiedlichen Strukturierungen führen können. Ein Vergleich der erzeugten Strukturierung mit einer manuell erstellten Klassifikation, die nur eine Sichtweise abbildet, reicht daher nicht.

Mithilfe von *Simulated Work Tasks* [Borlund, 2000] soll geprüft werden, ob der Benutzer die ihm gestellten Aufgaben lösen kann (Effektivität), wie viel Zeit er dafür benötigt (Effizienz) und wie zufrieden er ist (Zufriedenheit), also ob er damit insgesamt bei der Suche profitiert.

Literatur

[Belkin, 1993] Nicholas J. Belkin. Interaction with texts: Information retrieval as information-seeking behavior. In *Information Retrieval*, pages 55–66, 1993.

[Borlund, 2000] Pia Borlund. Experimental components for the evaluation of interactive information retrieval systems. *Journal of Documentation*, 56:71–90, 2000.

[Cool and Belkin, 2002] Colleen Cool and Nicholas J. Belkin. A classification of interactions with information. In H. Bruce, R. Fidel, P. Ingwersen, and P. Vakkari, editors, *Emerging frameworks and methods. Proceedings of the 4th COLIS*, pages 1–15, Greenwood Village, 2002. Libraries Unlimited.

[Fuhr *et al.*, 2010] Norbert Fuhr, Marc Lechtenfeld, Benno Stein, and Tim Gollub. The Optimum Clustering Framework: Implementing the Cluster Hypothesis. 2010. Submitted.

[Pang *et al.*, 2002] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up? Sentiment classification using machine learning techniques. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 79–86, 2002.

Semiautomatische Konstruktion von Trainingsdaten für historische Dokumente

Andrea Ernst-Gerlach, Norbert Fuhr
Abteilung Informatik und Angewandte Kognitionswissenschaft
Fakultät für Ingenieurwissenschaften
Universität Duisburg-Essen
47048 Duisburg, Deutschland

Abstract

Für Retrieval in historischen Dokumenten wird eine Abbildung der Suchbegriffe auf die historischen Varianten in den Dokumenten benötigt. Für diese Abbildung wurde ein regelbasierter Ansatz entwickelt. Der Engpass dieses Ansatzes ist die Konstruktion der Trainingsdaten. Dabei muss ein Experte manuell den historischen Formen, die dem Spellchecker unbekannt sind, die aktuelle moderne Form zuordnen. Zur Verbesserung dieses Verfahrens werden nun die Vorschläge des Spellcheckers betrachtet. Aus jedem Vorschlag und dem zugehörigen unbekannten Wort wird ein Beleg gebildet. Aus diesen Belegen werden nun wie gewohnt Regeln generiert und die häufigsten Regeln akzeptiert. Experimentelle Ergebnisse basierend auf der bisherigen Belegkollektion zeigen, dass ein großer Teil der Regeln auf diese Weise generiert werden kann. Dadurch können die Trainingsdaten deutlich schneller und mit geringerem manuellem Aufwand erzeugt werden.

1 Einleitung

Die Anzahl der digitalen historischen Kollektionen steigt kontinuierlich. Aber bei verfügbarer Volltextsuche für die Kollektionen werden viele Dokumente nicht gefunden, weil sie eine nicht-standardisierte Rechtschreibung verwenden. In vielen Ländern war die Schreibweise über viele Jahrhunderte hinweg nicht festgelegt. So wurde z. B. die deutsche Sprache erst 1901/1902 standardisiert [Keller, 1986]. Davor galt das Prinzip "Schreibe wie Du sprichst" (phonologisches Prinzip der Rechtschreibung). Z. B. ist $akzeptieren$ die moderne Form der Schreibvariante $acceptieren$.

Die nicht-standardisierte Schreibweise führt zu Fehlern, wenn in historischen Teilen von digitalen Bibliotheken gesucht wird. Die meisten Benutzer geben den Suchbegriff in moderner Sprache ein, die sich von der historischen Sprache in den Dokumenten unterscheidet.

Auch populäre Digitalisierungsinitiativen wie Google Book Search[1] oder die europäische digitale Bibliothek[2] unterstützen bisher keine Suche nach Schreibvarianten. Um dieses Problem zu lösen, arbeitet unser Projekt an der Entwicklung einer Suchmaschine, bei der der Benutzer seine Anfragen in aktueller Schreibweise eingeben kann, wenn er in historischen Dokumenten suchen möchte (siehe [Ernst-Gerlach and Fuhr, 2007]).

Andere Ansätze benutzen dafür wörterbuchbasierte Methoden (z. B. [Hauser *et al.*, 2007]). Allerdings hat diese Vorgehensweise den entscheidenden Nachteil, dass nur Wörter gefunden werden, die im Wörterbuch enthalten sind. Außerdem ist der zeitliche Aufwand für den manuellen Aufbau der Wörterbücher relativ hoch.

Die entwickelte Suchmaschine überwindet diesen Nachteil mit einem regelbasierten Ansatz, um das gesamte Vokabular abzudecken und dadurch den Recall zu erhöhen. Dafür werden Transformationsregeln entwickelt, die aus einem Suchbegriff die historischen Varianten generieren.

Durch die Orts- und Zeitabhängigkeit der Regeln müssen die Regelsätze jeweils neu generiert werden, wenn ein neues Korpus verfügbar wird. Diese Arbeit muss ohne Hilfe von Informatikern z. B. von Linguisten und Historikern geleistet werden. Deswegen ist es notwendig, ein Werkzeug zu entwickeln, das den Benutzer auf leicht verständliche und schnelle Art bei der Regelgenerierung unterstützt und dessen Benutzung keine Informatikkenntnisse voraussetzt.

Im Folgenden gehen wir davon aus, dass der Benutzer für eine neue Kollektion eine Volltextsuche ermöglichen möchte. Weil für Ort und Zeit der Kollektion kein Regelsatz vorhanden ist, muss der Benutzer zunächst Belege sammeln. Ein Beleg besteht aus der Flexionsform des Lemmas (im Folgenden Wortform genannt) und der zugehörigen historischen Variante. Erst im zweiten Schritt können die Regeln generiert werden.

Um die verschiedenen Benutzerinteressen darzustellen, wurden zwei Szenarios von Benutzertypen entworfen. Die Benutzertypen sollen dabei in erster Linie die Spannweite des notwendigen Bedarfs an Unterstützung bei der Erstellung von Belegen und Regeln darstellen. Z. B. könnte für einen Linguisten bereits die Bildung der Belege ein interessantes Forschungsthema sein. Er möchte die Belege nur mit semi-automatischer Unterstützung generieren, weil er auch an der Entwicklung der Sprache sowie an Regeln mit einer sehr hohen Precision interessiert ist. Er sucht oft nach allen Vorkommen eines Wortes in einer Kollektion und kann deswegen nur mit einem kompletten Regelsatz arbeiten. Im Gegensatz dazu möchte ein Historiker lediglich relevante Dokumente zu einem bestimmten Thema finden. Deswegen möchte er möglichst schnell eine Volltextsuche nutzen. Demzufolge wird er einen automatischen Ansatz bevorzugen. Auch wenn er deswegen zunächst einige Dokumente nicht findet, reicht es ihm aus, falls er die Möglichkeit hat, den Regelsatz später zu überarbeiten.

Ausgehend von seinen Bedürfnissen wird sich der Benutzer mehr auf den Recall oder die Precision seiner Suche konzentrieren. Das Tool soll an dieser Stelle die not-

[1] http://books.google.com/ g. a. 27.08.2010
[2] http://www.europeana.eu/portal/ g. a. 27.08.2010

wendige Flexibilität bieten. Dabei wird dem Benutzer Unterstützung für den gesamten Prozess der Regelgenerierung angeboten. Allerdings bleibt es ihm selbst überlassen, wie viele der vorgeschlagenen Belege (und Regeln) er akzeptiert.

Der vorliegende Artikel hat die folgende Struktur: Zunächst wird ein kurzer Überblick über die verwandten Arbeiten im Bereich der Erstellung von Trainingsdaten gegeben. Anschließend wird in Abschnitt 3 die Regelgenerierung erläutert. Abschnitt 4 zeigt, wie der Algorithmus zur Regelgenerierung auch zur automatischen Generierung von Belegen und Regeln verwendet werden kann. Der Ansatz wird in Abschnitt 5 evaluiert. Der letzte Abschnitt fasst den Artikel zusammen und gibt einen Ausblick auf zukünftige Arbeiten.

2 Verwandte Arbeiten

Gotscharek et. al. [Gotscharek *et al.*, 2009] haben mit dem LeXtractor ein Werkzeug zur Konstruktion von historischen Lexika entwickelt. Die Lexikoneinträge können auch als Belege für unseren Ansatz aufgefasst werden. Das Werkzeug hat zwei Ansichten. Die erste bietet dem Benutzer einen Text mit hervorgehobenen unbekannten Termen. In dieser Ansicht kann der Benutzer auf Basis des Textes Lexikoneinträge erzeugen. Die zweite Ansicht zeigt eine Liste mit unbekannten Termen, die nach absteigender Termhäufigkeit geordnet ist. Durch die Arbeit mit dieser Liste kann der Prozentsatz der vom Lexikon abgedeckten Wörter schnell gesteigert werden.

Da das Werkzeug zur Lexikonerstellung verwendet wird, müssen die Ergebnisse eine hohe Präzision aufweisen. Deswegen muss ein Experte alle unbekannten Wörter der ganzen Kollektion bearbeiten und jede Schreibweise beurteilen. Zur Unterstützung wird eine Liste mit Textstellen angeboten, wenn ein Wort für die Konstruktion eines Lexikoneintrags ausgewählt wird. Der LeXtractor verwendet manuell erstellte Regeln (sog. Patterns), um potenzielle moderne Formen in einem aktuellen Lexikon zu finden. Während der Konstruktion von Lexikoneinträgen ist es auch möglich, zusätzliche Patterns vorzuschlagen, wenn eine neue Regel bemerkt wird.

Pilz und Luther [Pilz and Luther, 2009] haben eine Methode entwickelt, um die Sammlung von Belegen in ihrem Evidencer Werkzeug zu unterstützen. Die Methode soll vor allem den Arbeitsaufwand für diesen Schritt verringern. Der Evidencer benutzt dazu einen Bayes'scher Klassifizierer. Dabei nehmen sie an, dass sich die Verteilung der N-gramme signifikant zwischen den Standard- und den Nicht-Standardschreibungen unterscheidet. Zur Einteilung in richtige Schreibweisen und Schreibvarianten schätzt der Klassifizierer die Wahrscheinlichkeit, ob es sich um eine Schreibvariante handelt. Um diese Wahrscheinlichkeit abzuschätzen, werden Trainingsbeispiele benötigt. Nach der Trainingsphase wird eine Liste mit unbekannten Wörtern präsentiert. Diese sortiert die Wörter absteigend nach der Wahrscheinlichkeit für Schreibvarianten. Der Benutzer kann den Klassifizierer anpassen, indem er den Grenzwert für mögliche Varianten verändert.

VARD 2 [Baron and Rayson, 2008] ist ebenfalls in der Lage, moderne Formen für Schreibvarianten in historischen Dokumenten zu finden. Das Werkzeug markiert alle potenziellen Varianten, die nicht in einem modernen Lexikon zu finden sind. Für jedes markierte Wort wird dem Benutzer eine Liste mit potenziellen zugehörigen modernen Schreibungen angeboten. Der Benutzer kann aus der Liste dann die passende moderne Form auswählen. Ein zweiter Modus bietet zudem die Möglichkeit, automatisch die Vorschläge mit dem höchsten Ranking zu akzeptieren, wenn der Wert über einem vom Benutzer festgelegten Mindestwert liegt. Um diese Vorschläge zu generieren, werden die folgenden drei Methoden benutzt:

- manuelle Liste von Beispielen für moderne Wörter und die zugehörigen Schreibvarianten,
- modifizierte Version des SoundEx-Algorithmus,
- manuell erstellte Liste von Ersetzungsregeln.

Basierend auf diesen drei Methoden wird ein Konfidenzwert für einen Vorschlag berechnet. Der Konfidenzwert ist dabei kein fester Wert, sondern wird nach jedem Schritt automatisch angepasst.

Der erste Ansatz benötigt (aufgrund seines Einsatzgebietes) ein großes Maß an manueller Interaktion sowohl für die Bildung der Belege als auch für die Bildung der Regeln. Der zweite Ansatz sieht mit Blick auf die automatische Unterstützung für den Benutzer vielversprechender aus. Der Benutzer hat hier die Möglichkeit, die Qualität der Ergebnisse durch einen Schwellwert zu beeinflussen. Allerdings benötigt der Klassifizierer eine große Anzahl an Trainingsdaten. Somit ist einiges an manueller Arbeit notwendig bevor der Klassifizierer eingesetzt werden kann. Außerdem kann der Benutzer nur dokumentweise vorgehen. Somit kann er nicht mehrere Vorkommen von möglichen Varianten in verschiedenen Texten gleichzeitig betrachten. Der dritte Ansatz sieht besonders wegen des ständig angepassten Konfidenzwertes für die modernen Formen sehr vielversprechend aus. Der Konfidenzwert ist ansonsten vergleichbar mit dem Bayes'scher Klassifizierer aus dem zweiten Ansatz. Der Nachteil von VARD 2 besteht in der Notwendigkeit von Trainingsdaten und Regeln als Eingabe. In beiden Fällen handelt es sich um manuell gesammelte Daten. Des Weiteren ist der SoundEx-Algorithmus ein phonetischer Algorithmus, der für die englische Sprache entwickelt wurde. Für die deutsche Sprache gibt es mit der Kölner Phonetik [Postel, 1969] ebenfalls einen phonetischen Algorithmus. Allerdings sind phonetische Algorithmen nur bedingt für die Erstellung von Regeln für historische Schreibweisen geeignet, weil sie durch die zeitliche und räumliche Veränderung der Aussprache ebenfalls angepasst werden müssten.

Zusammenfassend lässt sich feststellen, dass keiner der betrachteten Ansätze Belege automatisch generieren kann. Dadurch benötigen alle Werkzeuge einen hohen manuellen Aufwand, bevor sie einsatzbereit sind. Deswegen würde ein Werkzeug, das automatisch Belege für Trainingsmengen erzeugen kann, den Zugang zu historischen Dokumenten für den Benutzer deutlich erleichtern.

3 Generierung von Transformationsregeln

Im Folgenden werden unsere bisherigen Methoden zum Sammeln von Belegen und zur Regelgenerierung [Ernst-Gerlach and Fuhr, 2006] kurz erläutert. Zunächst werden Trainingsdaten benötigt, die moderne Wortformen auf zugehörige historische Varianten abbilden. Mit Hilfe einer Rechtschreibprüfung bekommen wir eine Liste von potenziellen historischen Schreibweisen. Zur Rechtschreibprüfung benutzen wir Hunspell[3], der zur Zeit Wörterbücher für 98 Sprachen zur Verfügung stellt. Die Vorschläge

[3] http://hunspell.sourceforge.net/ g. a. 27.08.2010

für falsch geschriebene Wörter basieren auf N-gramm-Vergleichen, Regeln und Aussprachedaten. Mit diesen Methoden werden dann auf Basis eines Wörterbuchs Vorschläge erstellt.

Durch eine manuelle Überprüfung wird festgestellt, ob es sich bei den unbekannten Wörtern wirklich um Schreibvarianten handelt. Ist dies der Fall, so werden die zugehörigen modernen Formen bestimmt. Zusätzlich benötigen wir noch die Termhäufigkeiten der historischen Formen. Anschließend können wir uns auf den nächsten Schritt konzentrieren — die Regelgenerierung.

Die automatische Regelgenerierung startet mit den erstellten Trainingsdaten. Dabei verwenden wir ein Tripel bestehend aus der modernen Wortform, der zugehörigen Schreibvariante sowie der Kollektionshäufigkeit der Schreibvariante.

Zunächst vergleichen wir jeweils die beiden Wörter und bestimmen sogenannte "Regelkerne". Diese beinhalten die notwendigen Transformationen und identifizieren den zugehörigen Kontext. Z. B. ergibt sich für die moderne Wortform *unnütz* und die historische Form *unnuts* die folgende Menge, die aus zwei Regelkernen besteht: (unn(ü→u)t), (t(z→s)).

In zweiten Schritt werden für jeden Regelkern die zugehörigen Regelkandidaten bestimmt. Diese berücksichtigen auch die Kontextinformationen (z. B. Konsonant (C) oder Wortende ($)) der modernen Schreibweise. Für das oben gezeigte Beispiel werden unter anderem die folgenden Regelkandidaten generiert: ü→ u, nü→ nu, üt→ ut, nüt→ nut, Cü→ Cu, z$→s$.

Im letzten Schritt werden die nützlichen Regeln durch Pruning der Regelmenge (wobei auch die Kollektionshäufigkeit berücksichtigt wird) durch eine modifizierte Version des PRISM Algorithmus (siehe [Cendrowska, 1987]) bestimmt. Dafür werden zunächst automatisch negative Belege erstellt und Precisionwerte für die Regeln berechnet. Anschließend werden die Regeln ausgewählt, die eine festgelegte Mindestprecision sowie eine Mindestvorkommenshäufigkeit aufweisen.

4 Automatisch akzeptierte Belege

Der letzte Abschnitt hat verdeutlicht, dass der bisherige Ansatz am Anfang einen hohen manuellen Aufwand benötigt. Deswegen ist es unser Hauptziel einen Algorithmus zu entwickeln, der Belege automatisch generieren kann, um diesen Aufwand zu reduzieren.

Die Annahme, dass Schreibvarianten ein bestimmtes Maß an Regularität beinhalten, bildet die Basis für den regelbasierten Ansatz. Basierend auf dieser Annahme sollen im Folgenden auch die Belege automatisch generiert werden. Die richtige moderne Form einer Schreibvariante befindet sich häufig unter den Vorschlägen des Spellcheckers. Wir nehmen an, dass diese Regularitäten zwischen moderner Form und Schreibvariante deutlich seltener auch zwischen Schreibvarianten und falschen Vorschlägen zu finden sind. Deswegen konzentriert sich unser Algorithmus (siehe Abbildung 1) auf das Problem, den richtigen Vorschlag des Spellcheckers zu einer Variante zu bestimmen. Dabei wird der Vorschlag ausgewählt, der über die häufigeren Regelkandidaten verfügt.

Aus jeder unbekannten Schreibweise und den zugehörigen Vorschlägen wird ein Beleg generiert (siehe Tabelle 1). Diese Belege bilden die Trainingsmenge aus der mögliche Regelkandidaten generiert werden. Da wir in diesem Schritt noch nicht die endgültigen Regeln generieren, sind hier die unterschiedlichen Regelkandidaten nicht relevant und es werden nur die Regelkerne betrachtet. Auf diese Weise erhalten wir eine eindeutigere Verteilung der Regeln.

Je häufiger ein Beleg in unterschiedlichen Belegen auftaucht, desto höher ist die Wahrscheinlichkeit, dass die Regel sinnvoll ist. Deswegen wird auch die Precision für Belege, die auf häufigeren Regelkernen basieren, höher sein. Demzufolge wird in jedem Durchlauf von den nicht akzeptierten Regelkandidaten derjenige mit der größten Häufigkeit akzeptiert (siehe Tabelle 2). Haben mehrere Regelkandidaten die gleiche Häufigkeit werden zunächst Substitutionsregeln akzeptiert, da diese meistens eine höhere Precision als Einfüge- und Löschregeln haben. Z. B. wird $i \rightarrow y$ gegenüber $s \rightarrow \emptyset$, $\emptyset \rightarrow h$ bevorzugt.

Regelkern	**Regelhäufigkeit**	**Entscheidung**
$i \rightarrow y$	7	akzeptieren
$un \rightarrow \emptyset$	1	
$un \rightarrow ge$	1	
$\emptyset \rightarrow ge$	1	
$w \rightarrow \emptyset$	1	
$ster \rightarrow ck$	1	
$ar \rightarrow zey$	1	
$l \rightarrow z$	1	
$i \rightarrow yt$	1	
$mann \rightarrow zeyt$	1	
$\emptyset \rightarrow je$	1	
$ig \rightarrow \emptyset$	1	

Tabelle 2: Beispiel für sortierte Regelkerne

Vorschlag	**Mögliche Variante**	**Entscheidung**
Geschicklichkeit	Geschicklichkeyt	akzeptieren
jederzeit	jederzeyt	akzeptieren
obgleich	obgleych	akzeptieren
Sonderheit	**In**sonderheyt	markiere $i \rightarrow y$ als akzeptiert

Tabelle 3: Beispiel zur Akzeptanz des Regelkerns $i \rightarrow y$

Nachdem wir einen Regelkandidaten akzeptiert haben, werden die zugehörigen Belege (und damit die Vorschläge der Rechtschreibprüfung) betrachtet. Basiert ein Beleg nur auf der akzeptierten Regel, wird er direkt akzeptiert. Basiert er auf mehreren Regeln, wird er akzeptiert, wenn alle anderen Regeln ebenfalls akzeptiert sind. Ansonsten wird lediglich markiert, dass der Regelkandidat akzeptiert ist (siehe Tabelle 3).

Nachdem wir nun akzeptierte Vorschläge haben, können wir im nächsten Schritt falsche Vorschläge aussortieren. Dafür nehmen wir an, dass zu jeder Schreibvariante nur eine moderne Schreibung existiert. Dadurch können wir die weiteren Vorschläge für historische Schreibweisen entfernen. Diese Annahme stellt eine Vereinfachung dar. Wie Pilz [Pilz, 2009] gezeigt hat, verfügt die Schreibvariante $Hunngern$ über die modernen Formen $Ungarn$ und $Hungern$. Die Vereinfachung ist an dieser Stelle notwendig, um die automatische Beleggenerierung überhaupt zu ermöglichen und somit den manuellen Aufwand für die Konstruktion der Trainingsdaten deutlich zu reduzieren. Diese Einschränkung gibt es bei der späteren ma-

```
Bilde Trainingsmenge aus Vorschlägen
Generiere Regelkandidaten (nur Regelkerne)
Solange Regelkandidaten r_j mit Regelhäufigkeit > min Regelhäufigkeit existieren
   Sortiere Regelkandidaten nach Häufigkeit
   Akzeptiere den häufigsten nicht akzeptierten Regelkandidaten r_i
    Markiere den Regelkandidaten r_i für alle zugehörigen Vorschläge s_i als
    markiert
    Akzeptiere alle s_i bei denen alle Regelkandidaten akzeptiert wurden
    Wenn s_i akzeptiert ist lösche alle konkurrierenden Vorschläge s_k
```

Abbildung 1: Algorithmus zur automatischen Beleggenerierung

Vorschlag	**Mögliche Varianten**	**Regelkandidaten**
Geschicklichke**it**	Geschicklichke**yt**	$i \rightarrow y$
Ungeschicklichke**it**	Geschicklichke**yt**	$un \rightarrow \emptyset, i \rightarrow y$
Unschicklichke**it**	**Ge**schicklichke**yt**	$un \rightarrow ge, i \rightarrow y$
Schicklichke**it**	**Ge**schicklichke**yt**	$\emptyset \rightarrow ge, i \rightarrow y$
Gesch**wister**lichke**it**	Gesch**ick**lichke**yt**	$w \rightarrow \emptyset, ster \rightarrow ck, i \rightarrow y$
jederzei**t**	jederzey**t**	$i \rightarrow y$
jeder**ar**t	jeder**zey**t	$ar \rightarrow zey$
jeder**l**ei	jeder**z**eyt	$l \rightarrow z, i \rightarrow yt$
jeder**mann**	jeder**zeyt**	$mann \rightarrow zeyt$
derzei**tig**	**je**derzey**t**	$\emptyset \rightarrow je, i \rightarrow y, ig \rightarrow \emptyset$

Tabelle 1: Beispiel für Trainingsdaten und generierte Regeln

nuellen Bearbeitung von Belegen nicht. Außerdem gehen wir davon aus, dass die Regeln, die durch diese Annahme verloren gehen, durch andere Belege mitgeneriert werden. Während des Löschungsprozesses werden die falschen Belege auch von weiteren zugehörigen Regelkernen entfernt. Anschließend startet der Prozess wieder mit dem häufigsten unbehandelten Regelkandidaten.

Der Benutzer kann diesen Prozess durch folgende Parameter beeinflussen:

- Minimale Wortlänge: Rechtschreibprogramme generieren für kurze Wörter meistens mehr Vorschläge als für lange Wörter. Zusätzlich ist die Wahrscheinlichkeit eine falsche historische Form zu generieren, deutlich höher, weil sich kurze Wörter mit einer größeren Wahrscheinlichkeit ähneln als lange Wörter. Deswegen kann die Precision für die automatischen Belege durch eine minimale Wortlänge erhöht werden.
- Minimale Anzahl an Regelvorkommen: Der Kern unseres Ansatzes besteht darin, dass die Regelhäufigkeit ein Indikator für die Precision ist. Deswegen ist die Regelhäufigkeit ein offensichtlicher Parameter. Als untere Grenze muss darüber hinaus eine Regel mindestens zweimal vorkommen.
- Maximale Anzahl der Regelanwendungen pro Wort: Je mehr Regelanwendungen benötigt werden, um ein modernes Wort auf eine potenzielle Variante abzubilden, desto unwahrscheinlicher ist es, dass es sich um eine Schreibvariante handelt. Insbesondere kurze Wörter können sehr leicht auf komplett andere Wörter abgebildet werden. Z. B. kann durch drei Regelanwendungen auf *derzeitig* das Wort *jederzeyt* generiert werden (siehe Tabelle 1).

Aus dieser Parameterauswahl bevorzugt der Historiker möglicherweise eine kürzere Wortlänge, eine geringere Anzahl an Regelvorkommen sowie eine höhere maximale Anzahl an Regelanwendungen, um einen hohen Recall zu erreichen. Auf diese Weise kann er, sofern er möchte, direkt mit der Suche auf der Kollektion beginnen. Im Gegensatz dazu wird der Linguist eventuell genau die umgekehrte Parameterauswahl treffen.

Der vorgestellte Ansatz wurde bereits in den RuleGenerator integriert. Der RuleGenerator ist ein interaktives Werkzeug und bietet eine graphische Benutzeroberfläche mit der der Benutzer Belege sammeln (siehe [Awakian, 2010]) und Regeln generieren kann (siehe [Korbar, 2010]). Die Ergebnisse des Prozesses zur automatischen Regelgenerierung werden dem Benutzer in einer graphischen Benutzeroberfläche in Form einer Liste angezeigt (siehe Abbildung 1). Darin werden zur Zeit Tripel aus moderner Wortform, Schreibvariante und den zugehörigen Regeln dargestellt. Der Benutzer kann anschließend einzelne Belege oder alle Belege komplett akzeptieren. Zusätzlich bietet die Liste Zugang zu den Textstellen in denen die Varianten vorkommen. Dadurch kann der Benutzer die Wörter auch in dem Kontext betrachten, wenn der Bedarf besteht. Im Anschluss an die Bearbeitung der automatischen Belege kann der Benutzer aus den noch nicht zugeordneten unbekannten Wörtern weitere Belege bilden.

5 Evaluierung

Als Testkollektion wurde unsere Belegedatenbank gewählt. Sie wurde basierend auf Texten der Nietzsche Rezeption[4] sowie weiteren kleineren Kollektionen[5] aufgebaut. Die

[4] `http://www2.inf.uni-due.de/Studienprojekte/Nietzsche/pp2001/die_cd/die_cd.htm` g. a. 27.08.2010

[5] Digitales Archiv Hessen-Darmstadt `http://www.digada.de/index.html` g. a. 27.08.2010, Bibliotheca Augustana. FH Augsburg. `http://www.hs-augsburg.de/~harsch/augustana.html`, g. a. 27.08.2010, documentArchiv.de `http://www.documentarchiv.de` g.a. 27.08.2010

	Anzahl Belege									
	1000	2000	3000	4000	5000	6000	7000	8000	9000	10000
Precision	0,28	0,25	0,25	0,24	0,22	0,19	0,18	0,18	0,17	0,17
Recall Regel	0,55	0,47	0,50	0,47	0,50	0,50	0,51	0,51	0,50	0,49
Recall Regelvorkommen	0,91	0,90	0,92	0,92	0,94	0,95	0,95	0,95	0,96	0,96

Tabelle 4: Recall und Precision für Regelkerne auf Basis Testdaten des jeweiligen Durchlaufs

		Anzahl Belege									
		1000	2000	3000	4000	5000	6000	7000	8000	9000	10000
Regel	Automatisch	0,14	0,24	0,31	0,36	0,39	0,41	0,43	0,46	0,47	0,49
	Manuell	0,26	0,51	0,63	0,77	0,80	0,82	0,84	0,90	0,94	1
Regelvorkommen	Automatisch	0,10	0,20	0,30	0,39	0,48	0,58	0,68	0,77	0,86	0,96
	Manuell	0,10	0,22	0,32	0,42	0,52	0,61	0,71	0,80	0,90	1

Tabelle 5: Recall für Regelkerne auf Basis Gesamttestdaten

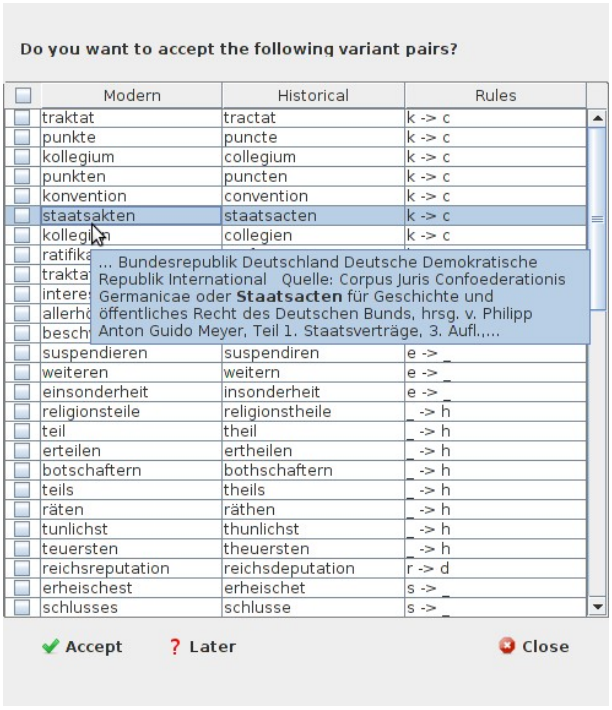


Abbildung 2: Benutzeroberfläche für automatische Belege

Texte stammen überwiegend aus dem 16. bis 19. Jahrhundert. An diesem Beispiel soll nachvollzogen werden, wie viel manueller Aufwand bei der Erstellung einer Trainingskollektion gespart werden kann. Um den sukzessiven Aufbau der Kollektion nachzustellen, wurde die Anzahl der als Testdaten verwendeten Belege von 1000 schrittweise um jeweils 1000 bis auf 10000 Belege erhöht. Für jede Testmenge wurde unser Verfahren angewendet. Als Parametereinstellung wurde mit einer Mindestwortlänge von fünf, mindestens zwei Regelvorkommen und maximal zwei Regelanwendungen pro Wort eine recallorientierte Auswahl getroffen. Dies wird insbesondere durch den geringen Wert für die minimale Vorkommenshäufigkeit der Regeln deutlich.

Es wurden zunächst Recall und Precisionwerte für die Regelkerne berechnet (siehe Tabelle 4). Die Precision sinkt von 0,28 Punkten bei 1000 Belegen bis 0,17 bei 10000 Belegen. Dies lässt sich durch die deutliche Parameterwahl zu Gunsten des Recall erklären. Da eine Regel nur zweimal vorkommen musste, um akzeptiert zu werden, steigt mit steigender Anzahl an Testdaten auch die Wahrscheinlichkeit, dass ein falscher Regelkern in einem weiteren potenziellen Belegpaar vorkommt. Beim Recall sind dagegen keine Auswirkungen der steigenden Anzahl der Trainingsdaten zu bemerken. Er hat eine Spannbreite von 0,47 bis 0,55. Somit lässt sich ungefähr die Hälfte der Regelkerne automatisch generieren.

Da sich die einzelnen Regeln sehr stark in ihrer Anwendungshäufigkeit unterscheiden, wurde zusätzlich auch noch der Recall basierend auf der Vorkommenshäufigkeit der einzelnen Regeln berechnet. Hier zeigt sich deutlich, dass vor allem die besonders häufigen Regeln generiert werden, da immer mindestens 90 % der Regelkerne gefunden werden.

Um die Entwicklung der Regelabdeckung einschätzen zu können, wurde noch die Entwicklung der Recallwerte der Regelkerne bezogen auf die Gesamtmenge der Belege berechnet (siehe Tabelle 5). Als Vergleichbasis für unser Verfahren diente dabei der Recall der Regelkerne bei manueller Erstellung der Testkollektion. Der Benutzer muss 2000 Belege manuell betrachten, um denselben Recall zu erreichen wie mit dem automatischen Verfahren. Betrachtet man den manuellen Aufbau der Testkollektion bezogen auf die Regelhäufigkeit zeigt sich, dass der Benutzer sogar über 9000 Belege manuell erzeugen muss, um den Recall so zu erhöhen, wie er es mit den automatisch erzeugten Be-

	Anzahl Belege									
	1000	2000	3000	4000	5000	6000	7000	8000	9000	10000
Automatische Belegabdeckung	0,88	0,91	0,93	0,93	0,94	0,94	0,94	0,95	0,95	0,95
Manuelle Belegabdeckung	0,91	0,95	0,96	0,98	0,98	0,98	0,99	0,99	1	1

Tabelle 6: Recall Belegabdeckung

legen könnte.

Um festzustellen, wie viele Belege man mit den generierten Regelkernen wiederfinden könnte, wurde auch noch der Recall für die Belegabdeckung berechnet (siehe Tabelle 6). Dies geschah sowohl für die automatische als auch für die manuelle Vorgehensweise. Dabei zeigt sich, dass in beiden Fällen bereits mit den aus 1000 Belegen generierten Regeln fast 90% der Belege abgedeckt werden. Dadurch wird nochmals die Regelhaftigkeit von Schreibvarianten gezeigt. Der Benutzer muss in etwa 2000 Belege manuell bewerten, um denselben Recall wie beim automatischen Ansatz zu erzielen.

6 Fazit

In diesem Artikel wurde ein Ansatz zur automatischen Konstruktion von Belegen vorgestellt. Die Belege werden als Eingabe für den Prozess der Regelgenerierung benötigt, der Retrieval in Texten mit nicht-standardisierter Rechtschreibung ermöglicht. Der vorgestellte Ansatz bietet dem Benutzer die Möglichkeit mehrere Parameter einzustellen. Dadurch ist der Ansatz sehr flexibel, weil der Benutzer den Prozess der Beleggenerierung entsprechend seiner Erwartungen an Recall und Precision beeinflussen kann.

Die Evaluierung hat deutlich gezeigt, dass die häufigen Regelkerne ausgewählt werden. Wenn der Benutzer diese direkt akzeptiert, muss er sich nur noch die Wörter anschauen, bei denen es nicht möglich war, einen Vorschlag automatisch zuzuordnen. Im weiteren Verlauf des Projektes soll daran gearbeitet werden, die Liste der verblieben unbekannten Wörter nach sinkender Irregularität zu sortieren.

Da die automatische Belege bereits nach sinkender Häufigkeit sortiert sind, wird diese Liste demnächst anhand der Regeln sortiert. Dadurch bekommt der Benutzer schneller einen Überblick über mögliche Regeln. Außerdem lässt sich so neben veränderten Parametereinstellungen auch die relativ geringe Precison steigern, weil der Benutzer die automatischen Belege schneller bearbeiten kann.

Literatur

[Awakian, 2010] A. Awakian. Development of a user-interface for an interactive rule development. Master's thesis, University of Duisburg-Essen, 2010.

[Baron and Rayson, 2008] A. Baron and P. Rayson. Vard 2: A tool for dealing with spelling variation in historical corpora. In *Proceedings of the Postgraduate Conference in Corpus Linguistics*, 2008. Aston University, Birmingham.

[Cendrowska, 1987] J. Cendrowska. PRISM: An algorithm for inducing modular rules. *International Journal on Man-Machine Studies*, 27(4):349–370, 1987.

[Ernst-Gerlach and Fuhr, 2006] Andrea Ernst-Gerlach and Norbert Fuhr. Generating search term variants for text collections with historic spellings. volume 3936 of *Lecture Notes in Computer Science*, Heidelberg, 2006. Springer Verlag. ISBN 3540333479.

[Ernst-Gerlach and Fuhr, 2007] Andrea Ernst-Gerlach and Norbert Fuhr. Retrieval in text collections with historic spelling using linguistic and spelling variants. In *JCDL '07: Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries*, pages 333–341, New York, NY, USA, 2007. ACM.

[Gotscharek *et al.*, 2009] A. Gotscharek, A. Neumann, U. Reffle, Ch. Ringlstetter, and K. U. Schulz. Enabling information retrieval on historical document collections: the role of matching procedures and special lexica. In *AND '09: Proceedings of The Third Workshop on Analytics for Noisy Unstructured Text Data*, pages 69–76, New York, NY, USA, 2009. ACM.

[Hauser *et al.*, 2007] Andreas Hauser, Markus Heller, Elisabeth Leiss, Klaus U. Schulz, and Christiane Wanzeck. Information access to historical documents from the early new high german period. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-2007) Workshop on Analytics for Noisy Unstructured Text Data*, 2007. Hyderabad, India, January.

[Keller, 1986] R. Keller. *Die Deutsche Sprache und ihre historische Entwicklung.* Helmut Buske Verlage, Hamburg, Germany, 1986. ISBN 3875481046.

[Korbar, 2010] D. Korbar. Visualisation of rule structures and rule modification possibilities for texts with non-standard spelling. Master's thesis, University of Duisburg-Essen, 2010.

[Pilz and Luther, 2009] T. Pilz and W. Luther. Automated support for evidence retrieval in documents with non-standard orthography. In Susanne Winkler Sam Featherston, editor, *The Fruits of Empirical Linguistics Process*, volume 1, pages 211–228, 2009.

[Pilz, 2009] T. Pilz. *Nichtstandardisierte Rechtschreibung - Variationsmodellierung und rechnergestützte Variationsverarbeitung.* PhD thesis, University of Duisburg-Essen, 2009.

[Postel, 1969] Hans Joachim Postel. Die Kölner Phonetik. Ein Verfahren zur Identifizierung von Personennamen auf der Grundlage der Gestaltanalyse. *IBM-Nachrichten*, (19):925–931, 1969.

An Evaluation of Geographic and Temporal Search

Fredric Gey†, Noriko Kando‡, Ray Larson†
†University of California, Berkeley USA
‡National Institute of Informatics, Tokyo JAPAN

gey@berkeley.edu , ray@ischool.berkeley.edu , kando@nii.ac.jp,

Abstract

This paper summarizes parts of the NTCIR-GeoTime Task held in Tokyo June 15-18, 2010. This task was the first evaluation specifically of search with both Geographic and Temporal constraints, i.e. it combines geographic information retrieval (GIR) with time-based search to find specific events in a multilingual collection. We describe the data collections (Japanese and English news stories), topic development, research approaches, assessment results and lessons learned from the evaluation.

1 Introduction

Semantic search queries and factoid questions require semantic processing to deliver results beyond bag-of-words search. Geo-temporal search concerns search which has both geographic and temporal constraints. In particular the search for events or to answer questions about events contains, often, specificity of location (where) and specifity of time (when). A simple example might be "When and where did Rosa Parks die?" in which the user wishes to know a "specific" date (is it year or month-year?) and "specific" location (should it be city?) to answer the question. A more complex question "How long after the Sumatra earthquake did its tsunami hit Sri Lanka?" has geographic constraints and wishes to extract a somewhat specific temporal expression (e.g. "a few hours") from the document collection being searched. The above examples are taken directly from NTCIR-GeoTime, the first evaluation of geo-temporal search recently presented (mid-June 2010) at the eighth NTCIR Workshop in Tokyo. The results clearly demonstrated that semantic markup for geography and time outperformed traditional IR methodologies.

Cultural Geographic search is quite prevalent in many modern search venues. A great number of documents (web, news, and scientific) have a geographic focus. Geographic search allows for a unique user interface, the interactive map, which can be utilized not only to narrow the user's focus by geography, but also to highlight interesting events. Geographic information retrieval is concerned with the retrieval of thematically and geographically relevant information resources in response to a query of the form {<*theme* or *topic*, *spatial relationship*, *location*>}, e.g. ``Temples within 5 km. of Tokyo". [Larson 1996, Jones et al 2004]. It has been estimated that 22 percent of web searches are location based [Asadi et al 2005]. Systems that support GIR, such as geographic digital libraries, and location-aware web search engines, are based on a collection of georeferenced information resources and methods to spatially search these resources with geographic location as a key. Information resources are considered geo-referenced if they are spatially indexed by one or more regions on the surface of the Earth, where the specific locations of these regions are encoded either directly as spatial coordinates, i.e. geometrically, or indirectly by place name [Hill 2006]. However, in order for place names to support a spatial approach to GIR, they must be associated with a model of geographic space. There have been over six workshops [Purves and Clough 2010] on Geographic Information Retrieval (GIR) held in association with SIGIR, CIKM, ECDL or other conferences as well as workshops and conference tracks on location-based search, there has also been 4 years of evaluation of GIR within CLEF (the GeoCLEF and GikiCLEF tracks [Mandl et al 2008,Santos et al 2010]). But, until this track at NTCIR, Asian language geographic search had never been specifically evaluated, even though about half of the NTCIR-6 Cross-Language topics had a geographic component (usually a restriction to a particular country).

The temporal aspects of search have been largely ignored in the IR community, but not in the GIS and computational linguistics communities. There has been a special issue of ACM TALIP on 'Temporal Information Processing' [Mani, Pustejovsky and Sundheim 2004], as well as at least two workshops on "Temporal and Spatial Information Processing". Use of temporal information in web search and results presentation (hit clustering) was explored in [Alonso, Gertz and Baeza-Yates 2007]. The NTCIR-GeoTime task organizers wanted to utilize and incorporate past research on this aspect as part of the evaluation.

2 Data and Test Topics

Two news story collections were used for NTCIR-GeoTime, one Japanese and one English. The Japanese collection was Mainichi newspapers for 2002-2005, which had 377,941 documents. The English collection, consisted of 315,417 New York Times stories also for 2002-2005. Users of the NYT collection had to pay a fee of $50US to the Linguistic Data Consortium to prepare and mail the DVD with this collection. Details about these collections and their characteristics may be found in the

GeoTime Overview [Gey et al 2010]. The collections matched those used in other tasks in NTCIR-8 (Advanced Cross-Language Question Answering [Mitamura et al 2010] and Multilingual Opinion Tracking [Seki et al 2010]).

Using Wikipedia as the 'ground truth', the organizers created 25 topics in English , phrased as questions, from the annual notable events summary.[1] Each of the 25 topics was vetted to hit at least one relevant document in both languages – the non-Japanese-speaking organizers used Google-translate to translate the topic and run it against the Mainichi collection and translate and examine the top documents. The process of topic development is also discussed in the Overview [Gey et al 2010]. Four topics were of the form 'When and where did <person> die?' with one minor variation: GeoTime0007: *How old was Max Schmeling when he died, and where did he die?* Another question was looking for a fixed list – GeoTime0016: *When and where were the last three Winter Olympics held?* Another, similar question – GeoTime0021: *When and where were the 2010 Winter Olympics host city location announced?* was very difficult because it wanted to know where (Prague, Czech Republic at the 115th session, July 2, 2003) the IOC (International Olympic Committee) announced that Vancouver would host the 2010 Winter games. In the opinion of the organizers, the most difficult topic was expected to be GeoTime0025: *How long after the Sumatra earthquake did the tsunami hit Sri Lanka?* This did prove to be one of the difficult topics, but not necessarily the most difficult. Topics were formatted in XML structures containing a description field and a more extensive narrative field, in both English and Japanese, as in:

```
<TOPIC ID="GeoTime-0001">
<DESCRIPTION LANG="EN">When and where did Astrid Lindgren die?</DESCRIPTION>
<DESCRIPTION LANG="JA">いつ、どこでアストリッド・リンドグレーンは亡くなりましたか？</DESCRIPTION>
<NARRATIVE LANG="EN">The user wants to know when and in what city the children's author Astrid Lindgren died.</NARRATIVE>
<NARRATIVE LANG="JA">ユーザは、いつ、どの都市で、児童書作家のアストリッド・リンドグレーンが死亡したかを知りたいと思っている。</NARRATIVE>
</TOPIC>
```

The full set of topics may be found at:
http://metadata.berkeley.edu/NTCIR-GeoTime/topics.php

3 Evaluation and Results

An evaluation run consisted of a ranked list of up to 1000 documents for each topic. Relevance judging was done in a traditional manner on a pool of the top 100 documents retrieved from all runs with duplicates removed.

3.1 Teams Submitting Evaluation Runs

Six teams submitted runs for the English collection and five registered teams ran the 25 topics against the Japanese collection (three other groups agreed to submit runs to broaden the pool – two of these groups are labeled 'anonymous' below).

[1] e.g. http://en.wikipedia.org/wiki/2002

Team Name	Organization submitting English runs
BRKLY	University of California, Berkeley
DCU	Dublin City University, Ireland
IITH†	International Institute of Technology, Hyderabad
INESC	National Institute of Electroniques and Computer Systems, Lisbon, Portugal
UIOWA	University of Iowa
XLDB	University of Lisbon, Portugal

† Run submitted late, not included in pooling

Table 1: Groups Submitting English Runs

Team Name	Organization submitting Japanese runs
Anon1	Anonymous submission 1
BRKLY	University of California, Berkeley
FORST	Yokohama National University, Japan
HU-KB	Hokkaido University, Japan
KOLIS	Keio University, Japan
Anon2	Anonymous submission 2
M	National Institute of Materials Science, Japan
OKSAT	Osaka Kyoiku University, Japan

Table 2: Groups Submitting Japanese Runs

The English groups submitted a total of 25 runs (a maximum of 5 different runs per team were allowed) and the Japanese groups submitted 34 distinct runs.

3.2 Results

Results in [Gey et al 2010] are displayed using three relatively well-established evaluation measures: Average Precision (AP), Q Measure, and Normalized Discounted Cumulative Gain (nDCG). Details about these evaluation measures which were also used for the IR4QA (Information Retrieval for Question-Answering) task of NTCIR-8 may be found in [Sakai et al 2010]. For simplicity we only display the nDCG results in the following table to show relative performance. A run is specified by team-name-topic-language-document-language-run_number-D or DN where D means description only which DN means description and narrative were used from the topic (the IIIT submission did not specify which fields were used)

RUN	nDCG
INESC-EN-EN-05-DN	**0.6246**
UIOWA-EN-EN-01-D	**0.6228**
BRKLY-JA-EN-01-DN	**0.617**
XLDB-EN-EN-02-T-DN	**0.5705**
DCU-EN-EN-02-D	**0.5513‡**
IIIT-H-EN-EN	**0.2224**

‡statistically significant difference (α=0.01) from the value of the run in the next row

Table 3: Best GeoTime English Run per Team

The most interesting result from this table is that Berkeley had better cross-lingual performance than its monolingual runs. This phenomenon appears occasion-

ally in Cross-Language Information Retrieval when blind feedback obtains additional discriminating terms from the top retrieved documents of an initial retrieval (BRKLY used blind feedback as a baseline [Larson 2010] without geotemporal extensions)..

Another way to compare performance is to fix the run type, for example to compare runs which all teams used only the D (description) part of the topic in their runs. The following table compares description only runs against the Japanese collection .

RUN	nDCG
HU-KB-JA-JA-03-D	0.5881†
KOLIS-JA-JA-04-D	0.5159†
Anon2-EN-JA-01-T	0.4231
M-JA-JA-03-D	0.3982
FORST-JA-JA-04-D	0.3772
OKSAT-JA-JA-01-D	0.3138
BRKLY-JA-JA-02-D	0.3014
Anon-JA-JA-02-UNK	0.2085

Table 4: Best Japanese D Run per Team (nDCG)
† statistically significant difference (α=0.05) from the value of the run in the next row

The interesting thing to immediately observe is that BRKLY which did so well in English runs comes in at a relatively low performance using the same blind feedback methodology as for English. Indeed, if we further exclude the anonymous runs (including M) for which we have no methodology , Berkeley's performance is worst among official Japanese runs. The reason for this has yet to emerge, however, all the other Japanese groups except OKSAT utilized sophisticated geotemporal processing in their approaches to retrieval.

4 Technical Approaches to Geo-Temporal Retrieval

In this section we review the technical approaches taken by the best performing teams.

4.1 English Approaches

A wide variety of approaches were utilized by the different groups. The most conventional was BRKLY's baseline approach of only doing probabilistic ranking coupled with blind relevance feedback. This worked very well for English, but for Japanese it substantially underperformed the approaches by other teams which submitted Japanese runs. Several groups (DCU from Dublin City University, Ireland, IIT-H of Hyderabad, India, and XLDB of University of Lisbon) primarily utilized geographic enhancements (although XLDB did consult DBpedia as an external resource using a timestamp) and did not perform as well as groups which tackled the temporal qualities of the retrieval.

A more elaborate approach was taken by the INESC group from Lisbon, Portugal who utilized a geographic resource (Yahoo PlaceMaker) for extracting geographic expressions and the TIMEXTAG[1] system from the University of Amsterdam for locating temporal expressions from within both topic and documents. Document processing was done at both the document and sentence level. Their hybrid approach relied upon the maximum amount of semantic content from the topic, so they utilized both description and narrative components from each topic. University of Iowa utilized a hybrid approach which combined probabilistic and (weighted) Boolean query formulation.

4.2 Japanese Approaches

The most straightforward of these geotemporal approaches was the KOLIS system of Keio University which merely counted the number of geographic and temporal expressions found in top-ranked documents of an initial search and then re-ranked based upon initial probability coupled with weighting of the counts. HU-KB of Hokkaido University , similarly to the University of Iowa for English, also combined probabilistic and Boolean query formulation [Mori 2010]. However, in the case of Hokkaido, the Boolean approach was utilized to filter out unwanted documents from the probabilistic ranking. In order to deal with the Boolean tendency to return the null set, HU-KB expanded the vocabulary using a synonym thesaurus. The FORST group of Yokahama University [Yoshioka 2010] used question decomposition to separate out temporal from locational aspects of the topics in order to apply standard factoid question-answering techniques which work well on a single question type (when or where). While KOLIS utilized a custom gazetteer of place names and a fixed list of temporal expressions (not including day-of-week), the Hokkaido [HU-KB] approach used the Cabocha system for named entity tagging [Kudo and Matsumoto 2002].

5 Topic Difficulty

There are two methods of assessing topic difficulty: looking at average performance over all runs by topic – the topics with low average precision are assumed to be the most difficult. The other way is to examine differences between median performance and maximum performance – this can demonstrate that particular methods perform better for such topics.

5.1 Topic Difficulty by Average Precision

Figures 1 and 2 average the three performance measures over all submitted runs and plot this average by topic.

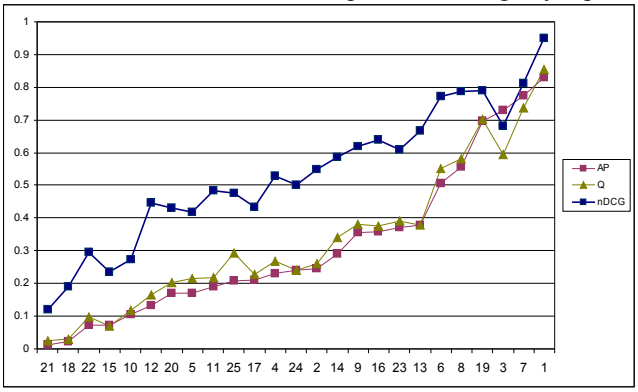


Figure 1: Per-topic AP, Q and nDCG averaged over 25 English runs for 25 topics (pool depth 100), sorted by topic difficulty (AP ascending)

[1] http://ilps.science.uva.nl/resources/timextag

The data are sorted by average precision in order to more clearly identify which topics presented the most challenge to successful search.

From the point of view of search of the English NYT collection, the four most difficult topics (less than 0.1 overall average precision) seem to be topic 15 (*What American football team won the Superbowl in 2002, and where was the game played?*), topic 18 (*What date was a country was invaded by the United States in 2002?*), topic 21 (*When and where were the 2010 Winter Olympics host city location announced?*) and topic 22 (*When and where did a massive earthquake occur in December 2003?*)

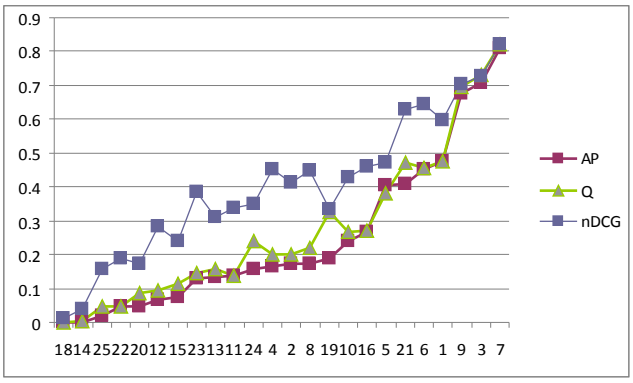


Figure 2: Per-topic AP, Q and nDCG averaged over 34 Japanese runs for 24 topics (pool depth 100), sorted by topic difficulty (AP ascending)

With respect to Japanese search of the Mainichi collection, several other topics (12, 14, and 25) also had average precision below 0.1 while topic 23 searches averaged 0.129. Topic 12 is *When and where did Yasser Arafat die?,* Topic 14 is *When and where did a volcano erupt in Africa during 2002?,* Topic 23 is *When did the largest expansion of the European Union take place, and which countries became members?,* and Topic 25, the one predicted by the organizers to be difficult: *How long after the Sumatra earthquake did the tsunami hit Sri Lanka?*

5.2 Median/Maximum Topic Peformance

Another way to assess performance is to examine individual performance variability across topics. Such performance can be displayed by taking individual topic runs and finding the minimum, median and maximum performance for that topic. These are displayed in Figures 3 (English runs) and 4 (Japanese runs). While for nearly all Japanese topics, at least one group had a minimum precision of near zero for that topic, there was still a wide variability of performance from both minimum to median average precision for a topic, as well as from median precision to maximum precision for a topic. Where the median and maximum are very close, we can infer that almost all groups had good performance.

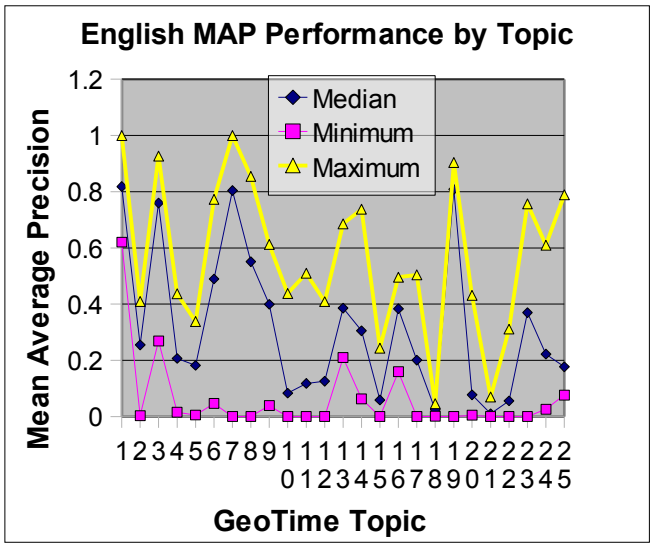


Figure 3: Per-topic AP showing Minimum, Median and Maximum performance for English runs

An example for English where median and maximum are almost identical is topic 19: *When and where did the funeral of Queen Elizabeth (the Queen Mother) take place?* An example where the best run (UIOWA-EN-03-DN, maximum AP 0.7889) is more than four times better than the median (0.177) is for topic 25: *How long after the Sumatra earthquake did the tsunami hit Sri Lanka?*

An example (for Japanese) where median and maximum are almost identical is topic 7: *How old was Max Schmeling when he died and where did he die?* On the other hand, topic 19, which showed almost no variation between median and maximum for English, becomes, for Japanese, an example where the maximum precision (1.000, run FORST-JA-JA-02-D) is more than 7 times better than the median precision (0.1339).

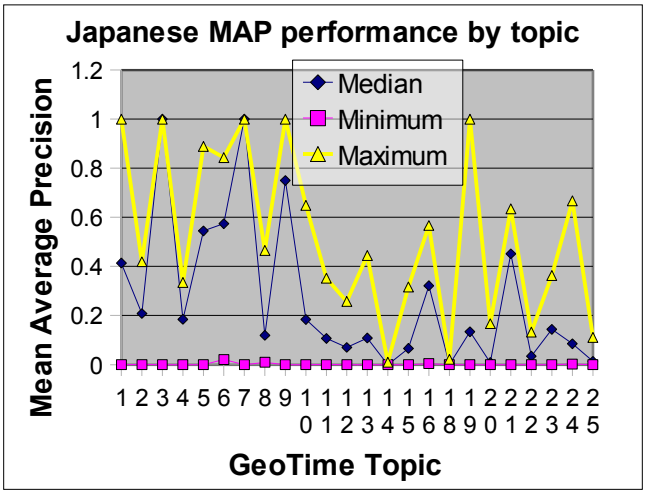


Figure 4: Per-topic AP showing Minimum, Median and Maximum performance for Japanese runs

It is worth noting that the minimum for Japanese was a single run in which the team did very poorly on all topics. It should probably be considered an outlier and removed from future analysis. The median performance is a more reliable statistic from which to draw conclusions.

6 Judgment Approaches for Imprecise Temporal Expressions

One of the difficulties in relevance judgment is how to approach the extreme variability in temporal expressions in text and how to approach judgment, particularly with respect to these expressions. As a point of reference, each document had a specific date upon which it was published. At least for English relevance judgments, imprecise expressions relating to that date were seen as sufficient evidence to judge a document relevant. For example if a document stated "Katherine Hepburn died Wednesday in her home in Connecticut" it was assumed that sophisticated natural language processing could infer the exact date of death from the date of the document. If a document stated (for topic 25) that "*a few hours later* the Sumatra earthquake tsunami hit the coast of Sri Lanka" the document could be judged relevant. Finally, we retrospectively realized that a topic needs to be **date stamped** if it asks a temporally relative question. For example topic 16: *When and where were the last three Winter Olympics held?* was formulated before the 2010 winter Olympics were held in Vancouver. Thus while documents could have known that the 2010 Winter Olympics were to be held in Vancouver, the correct answer (for a topic date-stamped before 2010) would be 1998 (Nagano, Japan), 2002 (Salt Lake City, USA), and 2006 (Turin, Italy).

7 Discussion

7.1 Lessons Learned

NTCIR-GeoTime was the first attempt at evaluating geotemporal information retrieval. While Geographic Information Retrieval has had numerous evaluations, the addition of a temporal component has proven very challenging to participants, especially if the topic (question) can be misinterpreted by the automated retrieval process (as in the case of topic 21: *When and where were the 2010 Winter Olympics host city location announced?*) or require a list answer which is time varying (topic 16: *When and where were the last three Winter Olympics held?*). Teams which relied exclusively on geographic enhancements did not perform as well as those which incorporated some temporal expression processing within their methodologies. Questions remain as to why there was so much performance variability across document collection language (Japanese and English) for the same topics.

7.2 Future Directions

Plans are already being formulated for a second GeoTime evaluation for the NTCIR-9 Workshop in 2011. We are exploring additional languages – Korean and Chinese to the document collection set. For participant groups we will make available a standard set of resources (gazetteers, named entity taggers, TimexTag, etc). In addition, we have a definite desire to evaluate location-based and map-based search simulation, i.e. "What event is happening "here" and "now/tomorrow"" -- where here and now come from the included latitude/longitude coordinates. This should facilitate innovative result visualization using Google/MS Earth/map as well as map-based querying (bounding rectangles).

8 Acknowledgments

We thank participant assessors Christopher Harris (University of Iowa), Krishna Janakiraman (UC Berkeley), Ricardo Vaz and Flávio Esteves (Technical University of Lisbon). Appreciation is also due to Jorge Machado (INESC, Lisbon, Portugal) who programmed the English assessment system, and Tetsuya Sakai (Microsoft Research, Asia) who ran the evaluation statistics. The work of the first author was supported by a Visiting Researcher travel grant from the National Institute of Informatics of Japan during June-July 2010. We thank the LWA/WIR reviewers for suggesting improvements and corrections to the original submission.

9 References

[Alonso, Gertz and Baeza-Yates 2007] On the Value of Temporal Information in Information Retrieval, SIGIR Forum, Vol. 41 No. 2 December 2007, pp 35-41.

[Asadi *et al*., 2005] S Asadi, , C.-Y. Chang, X. Zhou, and J. Diederich. Searching the world wide web for local services and facilities: A review on the patterns of location-based queries. In W. Fan, Z. Wu, and J. Yang, editors, WAIM2005, pp. 91–101. Springer LNCS 3739, 2005.

[Gey et al 2010] F. Gey, R. Larson, N. Kando, J. Machado and T. Sakai, NTCIR-GeoTime Overview: Evaluating Geographic and Temporal Search, In Proceedings of the 8th NTCIR Workshop Meeting , Tokyo Japan June 15-18, 2010, ISBN: 978-4-86049-053-9.

[Harris 2010] C. Harris, Geographic Information Retrieval Involving Temporal Components, in Proceedings of the 8th NTCIR Workshop Meeting.

[Hill 2006] L L Hill, *GeoReferencing: The Geographic Associations of Information*, MIT Press, Cambridge, MA 2006.

[Jones et al 2004] C. B. Jones, A. I. Abdelmoty, D. Finch, G. Fu, and S. Vaid. The SPIRIT spatial search engine: architecture, ontologies and spatial indexing. In GiScience 2004, Oct. 2004, Adelphi, MD, pages 125–139, 2004. Cunningham

[Kishida 2010] K. Kishida, Vocabulary-based Re-ranking for Geographic and Temporal Searching at NTCIR GeoTime Task, in Proceedings of the 8th NTCIR Workshop Meeting .

[Kudo and Matusomoto 2002] T. Kudo and Y. Matsumoto, Japanese Dependency Analysis using Cascaded Chunking, in CoNLL 2002, Taipei.

[Larson 1996] R. Larson, Geographic information retrieval and spatial browsing. In GIS and Libraries: Patrons, Maps and Spatial Information, pages 81–124. UIUC - GSLIS, Urbana-Champaign, IL, 1996.

[Larson 2010] R Larson, Text Retrieval Baseline for NTCIR-GeoTime, in Proceedings of the 8th NTCIR Workshop Meeting .

[Machado, Borbinha and Martins 2010] J. Machado, J. Borbinha and B. Martins, Experiments with Geo-Temporal Expressions Filtering and Query Expansion at Document and Phrase Context Resolution, In Proceedings of the 8th NTCIR Workshop Meeting .

[Mani, Pustejovsky and Sundheim 2004] I. Mani, J. Pustejovsky, and B. Sundheim. Introduction to the special issue on temporal information processing. ACM Transactions on Asian Language Information Processing (TALIP), 3(1):1–10, 2004

[Mandl et al 2008] T. Mandl, F. Gey, G. Di Nunzio, N. Ferro, M. Sanderson, D. Santos and C. Womser-Hacker, An Evaluation Resource for Geographic Information Retrieval, In Proceedings of the Sixth International Language Resources and Evaluation (LREC'08) Morocco, May, 2008

[Mitamura et al 2010] T Mitamura, H Shima, T Sakai, N Kando, T Mori, K Takeda, C-Y Lin, R Song, C-J Lin, C-W Lee, Overview of the NTCIR-8 ACLIA Tasks: Advanced Cross-Lingual Information Access, in Proceedings of the NTICIR Workshop 8, Tokyo Japan June 15-18, 2010.

[Mori 2010] T. Mori, A Method for GeoTime Information Retrieval based on Question Decomposition and Question Answering, in Proceedings of the 8th NTCIR Workshop Meeting.

[Purves and Clough 2010] R. Purves, C. Jones, and P. Clough. GIR'10: 6th workshop on geographic information retrieval, 2010. http://www.geo.unizh.ch/rsp/gir10/index.html.

[Santos et al 2010] D. Santos, L. Cabara, et al, GikiCLEF: Crosscultural Issues in Multilingual Information Access in Proceedings of LREC 2010, Malta, May 2010.

[Sakai et al 2010] T. Sakai, H. Shima, N. Kando, R. Song, C-J. Lin, T. Mitamura, M. Sugimito, C-W. Lee Overview of NTCIR-8 ACLIA IR4QA, in Proceedings of the 8th NTICIR Workshop Meeting, Tokyo Japan June 15-18, 2010.

[Seki et al 2010] Y. Seki, L-W Ku, L. Sun, H-H. Chen and N. Kando,, Overview of Multilingual Opinion Analysis Task at NTCIR-8: A Step Toward Cross Lingual Opinion Analysis, in Proceedings of the 8th NTICIR Workshop Meeting, Tokyo Japan June 15-18, 2010.

[Yoshioka 2010] M. Yoshioka, A Method for GeoTime Information Retrieval based on Question Decomposition and Question Answering, In Proceedings of the 8th NTICIR Workshop Meeting, Tokyo Japan June 15-18, 2010.

Image Retrieval on Mobile Devices

Adrian Hub
University of Bamberg
D-96052, Bamberg, Germany
adrian.hub@uni-bamberg.de

Abstract

With the growing number of mobile devices and the access possibility to thousands of images from these devices, the users call for efficient image search techniques for mobile devices. Desktop paradigms cannot be used with the smaller screen sizes, hence it is needful to offer alternative searching and browsing strategies, which are adapted for mobile devices. In this paper we describe our ideas how image retrieval on mobile devices can be accomplished.

1 Introduction

The amount of small mobile devices, which we use in our every day life, grows constantly. There are cell phones, smart phones, tablet PCs, netbooks and so on. Most of them are equipped with more or less powerful cameras and all of them offer enough storage capacity to take and store a lot of photos on the device itself.

Furthermore the Internet is easily accessible via broadband connections offering access to an unlimited number of images. There are image search services like Google images[1], photo communities like Flickr[2] or Picasa[3] and social networks like Facebook[4]. All of them let the user search for thousands of images and the latter ones let him especially browse through the photos of friends and colleagues.

Due to the close integration of social networks, the access to the Internet and the huge storage capacity on mobile devices, a lot of own and external (from the web) photos are stored on the device or at least accessible. Unfortunately, most mobile devices are usually not designed to manage thousands of photos concerning the small screen size and limited control possibilities. The typical thumbnail view is inapplicable, as if the device should give a good overview with thumbnails, they had to be tiny making recognition of images very difficult or if the recognition with bigger thumbnails could be good, there can be placed only few on the small screen of the device regarding their typical screen sizes less than 4 inches.

Another drawback could be the lack of a keyboard, which makes tagging and searching for tags quite hard. As our measurements with Flickr crawls showed, a lot of images remain untagged (about 40% in this 'classical environment'), so tagging on mobile devices will probably be even less used. One more reason why image retrieval on mobile devices is a challenging topic.

Hence, we presented a hybrid system, called Picadomo [Hub *et al.*, 2009], that makes use of Hierarchical Faceted Search (HFS) [Hearst *et al.*, 2002] for the purpose of image retrieval and is adapted for the small screen size of mobile devices. It is based on our visual faceted search for desktop PCs VisualFlamenco [Müller *et al.*, 2008] and combines a tag-based search, search techniques based on EXIF data as well as Content-Based Image Retrieval (e.g. low-level visual features) to generate a good browsing experience and help the user for finding desired images on mobile devices (see Fig. 1).

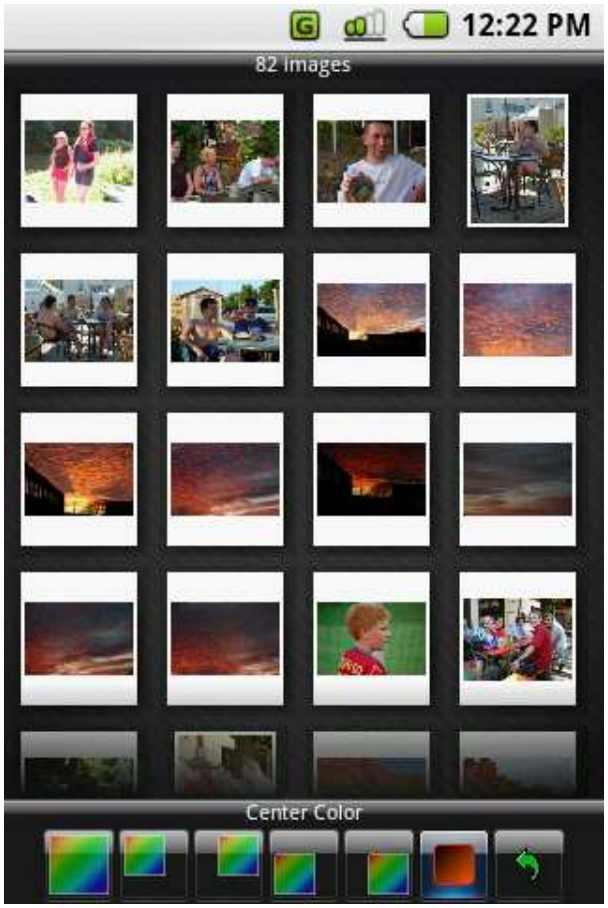


Figure 1: Screenshot of our Prototype

We now want to improve our prototype for mobile devices in different ways. First we want to address more image sources. Picadomo could handle only images stored on

[1] http://www.google.com/imghp
[2] http://www.flickr.com/
[3] http://picasaweb.google.com/
[4] http://www.facebook.com/

the internal memory card, due to the fact, that the feature extraction takes some time and cannot be done on the fly. Therefore it is essential to use server and/or peer-to-peer networks to broaden the searchable image sources.

Second, the facets used to search the images can be improved further. Our user experiments in [Hub *et al.*, 2009] showed, which facets were used often and which facets could be neglected. But it often depends on the user and/or the dataset, e.g. EXIF data like the time or place the picture was taken may be only interesting for own content, but not for images from the Internet.

Third, the search user interface can be enhanced in many ways. There are a lot of techniques in text retrieval, regarding the query specification, presentation of results, query reformulation, personalization and visualization, just to name a few. Some of them can be adopted for image retrieval as well, e.g. keyword-in-context (KWIC) views present extracted query terms along with other kinds of information (such as document title) about a specific search result. Similar to this, a selected color facet can be shown as overlay on image search results.

The paper is organized as follows: In the next section we briefly describe related work, in Section 3 we discuss different image sources along with their problems and methods of resolutions. Section 4 describes the two main search strategies the user typically is faced with. In Section 5 we discuss some user interface enhancements before we summarize our work in Section 6.

2 Related Work

Most of the existing applications on mobile devices browse images just by folder, some let the user assign tags to images or classify them in albums and few let the user browse multiple photo albums from Facebook, Picasa, Flickr and your memory card.

The *JustPictures* [Quillard, 2010] application allows the user to browse the above mentioned photo albums on the web, it shows EXIF data, it automatically notifies the user of album updates, can handle authentication of users to access private albums and many more (see Figure 2 for the album view of this application).

But the user has still to know where to look for an image, as there is no search interface. JustPictures shows EXIF data for photos, but there is no possibility to browse the photos by one selected EXIF feature.

Another way to organize personal photo collections on mobile devices is using time information as main ordering criterion for visualization and interaction [Harada *et al.*, 2004]. But time information should not be the only aspect, since browsing and searching by facets can offer much more possibilities.

A multi-faceted image search and browsing system, named *Scenique* [Bartolini, 2009], allows the user to manage photo collections by using both visual features and tags, possibly organized into multiple dimensions (see Figure 3). But this solution is not designed for mobile devices, as it was made for desktop-PCs and therefore requires a screen with higher resolution.

Within this section, we only describe some of the existing applications that deal with the administration of image collections on mobile devices. Mor Naaman et al. give a more detailed overview in [Naaman *et al.*, 2008].

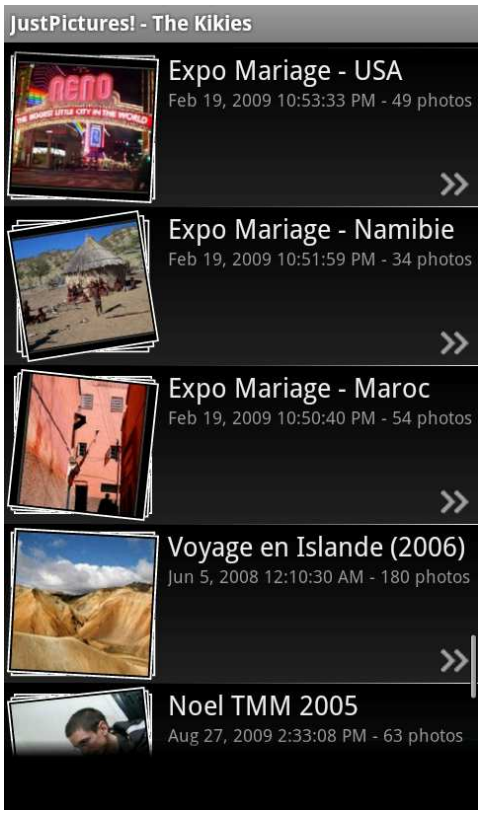


Figure 2: JustPictures in album view

3 Feature Database

Our approach relies on extracted features, that are stored in some kind of database and are accessible through the mobile device. Obviously, every image needs to be processed with our feature extractor to make it findable on the mobile device. This is no problem for images on the internal memory, as their features can just be stored locally on the mobile device, after they have been extracted. But this could get a little bit more challenging for external image sources, like the web or social network sites. Therefore we need to differentiate between local images, images from friends (e.g. Facebook or Flickr) and external sources (e.g. Google images or any web page).

3.1 Server Database

If we want to use our faceted search for searching images on the Internet, the best solution would probably be a server, that stores the facet data. The server can crawl the Internet and extract the features for given images. The database could then contain all facet data along with a resized version of the image and an URL, where the original image can be found. The mobile device obtains this information from the server to offer the results to the user.

Another possibility could be the cooperation with photo album websites. Whenever users upload their images, the features could be extracted and offered via an interface to the mobile devices, to make the images easily searchable.

Obviously, both solution require server power, for extracting and storing the facets. To avoid these costs a peer-to-peer (P2P) solution may come in handy.

3.2 Peer-to-Peer Database

Another approach to store the facet data is to share it in a peer-to-peer environment. All users of the application can

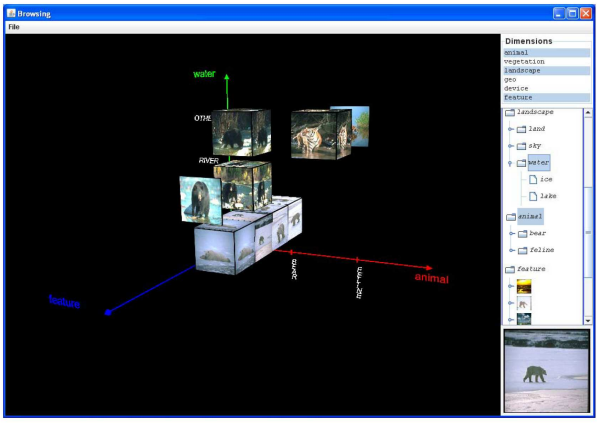


Figure 3: Scenique browsing interface

define their local photos as private, public or only viewable for friends. Depending on these settings and the relationship between the users they are able to browse public albums or photos of their friends using a P2P connection.

The process of feature extraction and the storage of facet data happens all on the mobile device. If a user starts the search for an image, the desired feature set will be exchanged using the P2P-network. Matching image results will then be forwarded to the searching user.

A JXTA-based implementation of a similar scenario was presented in [Müller *et al.*, 2007].

3.3 Local Database

If a user wants to use our faceted browsing application for browsing through her friend's photos, but her friends don't use the application or even don't have a mobile device, there is another possible solution. The user could simply mark some albums of her friends in our application as favorite albums, to indicate, that she wants to make these albums searchable via faceted browsing. Our application then processes these albums in the background, extracting the image features and stores them on the local memory. That way the user can search her own and the favorite albums of her friends with our faceted browsing application.

4 Search Strategies

When searching for images, there are generally two different use cases. First, recover images that the user has already seen before, e.g. photos taken with the camera of the mobile device, his own web albums or an image of a web album of any friend. Second, discover new images that the user hasen't seen before, e.g. images from the Internet or recently added photos of any web album of a friend. The search strategies for the two use cases may vary in detail.

4.1 Recover

Obviously the easier task is to recover an already seen image. The user may have a rough imagination of the image content, so we can use content-based image retrieval. As an example, the dominant color of a (region of an) image can help finding the image again. Other possibilities are the contrast of the image, texture or orientational features.

Any known metadata of an image can help finding it. EXIF features like the date or place the photo was taken, the camera model or if the photo was taken with or without flash allow simple browsing and searching. These facets can be used best for the user's own photos or for professional users.

If the user has his photos tagged, a tag based search can offer another possibility to search within photos. Due to the fact, that a lot of images remain untagged, this may be useful for only a small number of images.

Another option for searching images can be based on GPS data. If GPS data are available for a given set of photos, these photos can be placed on a map according to the place the photos were taken. In doing so, the user can easily search for photos of famous places via zooming and scrolling in a world map.

4.2 Discover

As a matter of principle, all approaches for recover images can also be used to discover unseen images. The expectation of the user regarding the resulting images is not as detailed given in this case. Hence, it depends on the individual use case, if it is an easier task to search for unseen images or not.

In addition the search for unseen images should support a key word query based on user input, as used for example on Google images. The result set of a key word query can then be further searched using our facets.

5 Search Interface

Regarding the search interface we want to describe in short how we could enhance the user interface and the presentation of results.

5.1 User interface

The search interface may be the most important part for searching and browsing through images. For a detailed illustration of our interface, see the prototype in [Hub *et al.*, 2009]. According to the user's feedback some improvements can be made. A lot of users asked for a timeline view to arrange the images chronologically. With individual time ranges this could indeed be a good way to search images, e.g. if the user wants to see the images of her last summer holiday, she can adjust the time range from August to September and only photos taken during that time will be shown.

Furthermore, our user experiments showed, that the number of color facets can be reduced. The fine grained color shades of our prototype were not used very often. We believe that ten to twelve main color shades are enough (see color picker at Google images). Other unused facets like the contrast for global an local regions can be reduced to only global contrast features.

Most of modern mobile devices are equipped with motion sensors, so it could be useful to use this for navigational purpose. We could imagine to scroll through a result list with pitching the mobile device. If the user shakes the device, the search could be reseted or the last step could be undone.

Alternatively multitouch gestures can be used for scrolling, zooming and navigation through search history.

5.2 Presentation of Search Results

Another very important aspect when searching for images is the presentation of search results. A simple list view is used most of the time, as usual with web search engines. Regarding text retrieval, theses lists can be enhanced with various improvements. There are summary information for every hit, query term highlighting, sparklines, preview of

document content and many more (see [Hearst, 2009] for more information).

Some of the well-established techniques for text retrieval can be adjusted and adapted for image retrieval. For example, the global dominant color of an image can be represented with a colored frame around the image dyed in the corresponding color. This can also be used as overlay for some regions of an image, to indicate the selected color feature similar to query term highlighting.

The context of an image can also easily displayed. If the result image comes from a webpage or an web album, the previous and next image can be presented as thumbnail along with the result image. This gives the user a quick and small overview of the result and she can recognize, if there are more similar images in one continuous photo stream or if it is a collection of totally different images, what could come in handy for image search on web pages.

A totally different way to present the search result could be the arrangement of the result set as an three dimensional image globe. The images can be arranged on a virtual globe according to their dominant color using the HSV color model [Zhang *et al.*, 1999]. The north pole is for light images with high values and the south pole for dark images with low values. The hues are grouped around the equator and all images of the result set are positioned where they fit best. The saturation can be neglected in this model. The navigation could be done easily by pitching the mobile device, to let the image globe roll. For big result sets, the user can zoom in to show more or bigger images.

6 Conclusion

In this paper we described some ideas how image retrieval on mobile devices can be improved. Our first prototype should be enhanced with the presented features. We believe that this is a useful approach for finding images from webpages, online albums and internal storage.

Besides the realization of mentioned features in our prototype the search for partners could be very helpful, to include the faceted image search into existing photo sharing communities.

References

[Bartolini, 2009] Ilaria Bartolini. A mulit-faceted browsing interface for digital photo collections. *CBMI*, 7:237–242, 2009.

[Harada *et al.*, 2004] Susumu Harada, Mor Naaman, Yee Jiun Song, QianYing Wang, and Andreas Paepcke. Lost in memories: interacting with photo collections on pdas. In *JCDL '04: Proceedings of the 4th ACM/IEEE-CS joint conference on Digital libraries*, pages 325–333, New York, NY, USA, 2004. ACM.

[Hearst *et al.*, 2002] Marti Hearst, Ame Elliott, Jennifer English, Rashmi Sinha, Kirsten Swearingen, and Ka-Ping Yee. Finding the Flow in Web Site Search. *Commun. ACM*, 45(9):42–49, 2002.

[Hearst, 2009] Marti A. Hearst. *Search User Interfaces*. Cambridge Univeristy Press, 2009.

[Hub *et al.*, 2009] Adrian Hub, Daniel Blank, Wolfgang Müller, and Andreas Henrich. Picadomo: Faceted image retrieval for mobile devices. *CBMI*, 7:249–254, 2009.

[Müller *et al.*, 2007] Wolfgang Müller, Soufyane El Allali, Daniel Blank, Andreas Henrich, and Thomas Lauterbach. Hunt the Cluster: A Scalable, Interactive Time Bayesian Image Browser for P2P Networks. *Multimedia Workshops, International Symposium on*, 0:317–322, 2007.

[Müller *et al.*, 2008] Wolfgang Müller, Markus Zech, Andreas Henrich, and Daniel Blank. VisualFlamenco: Dependable, Interactive Image Browsing Based on Visual Properties. CBMI International Workshop on Content-Based Multimedia Indexing, 2008.

[Naaman *et al.*, 2008] Mor Naaman, Rahul Nair, and Vlad Kaplun. Photos on the go: a mobile application case study. In *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 1739–1748, New York, NY, USA, 2008. ACM.

[Quillard, 2010] Gregory Quillard. Justpictures. http://code.google.com/p/justpictures/, 2010.

[Zhang *et al.*, 1999] L. Zhang, F. Lin, and B. Zhang. A CBIR Method Based on Color-Spatial Feature. *IEEE Region 10 Annual International Conference*, pages 166–169, 1999.

Named Entity Disambiguation for German News Articles

Andreas Lommatzsch, Danuta Ploch, Ernesto William De Luca, Sahin Albayrak
DAI-Labor, TU Berlin, 10587 Berlin, Ernst-Reuter-Platz 7
{andreas.lommatzsch, danuta.ploch, ernesto.deluca, sahin.albayrak}@dai-labor.de

Abstract

Named entity disambiguation has become an important research area providing the basis for improving search engine precision and for enabling semantic search. Current approaches for the named entity disambiguation are usually based on exploiting structured semantic and lingual resources (e.g. WordNet, DBpedia). Unfortunately, each of these resources cover independently from each other insufficient information for the task of named entity disambiguation. On the one hand WordNet comprises a relative small number of named entities while on the other hand DBpedia provides only little context for named entities. Our approach is based on the use of multi-lingual Wikipedia data. We show how the combination of multi-lingual resources can be used for named entity disambiguation. Based on a German and an English document corpus, we evaluate various similarity measures and algorithms for extracting data for named entity disambiguation. We show that the intelligent filtering of context data and the combination of multi-lingual information provides high quality named entity disambiguation results.

1 Introduction

Named entity recognition (NER) and named entity disambiguation (NED) are usually seen as a subtask of information extraction aiming to identify and classify text elements into predefined categories, such as name of persons, organizations, or locations. Ambiguous names are resolved by determining the correct referent. Data provided by NER and NED is required for various applications reaching from semantic search and clustering to automatic summarization and translation. The process of named entity recognition and disambiguation usually consists of three steps:

1. Identify words (or groups of words) representing an entity in a given text
2. Collect data for describing the identified entity in detail (entity properties, the relationship to other entities)
3. Classify the identified entity and calculate which candidate entity (from an external knowledge base) matches best (for resolving ambiguity).

The most complex step in the disambiguation process is the collection of metadata for the identified entity describing the entity context. Many systems use linguistic grammar-based techniques as well as statistical models for NED. Manually created grammar-based systems often obtain a high precision, but show a lower recall and require months of work by linguists. Other systems are based on deploying dictionaries and lexical resources, such as WordNet [Fellbaum1998] or DBpedia [Lehmann *et al.*2009].

In this paper we evaluate how comprehensive, multilingual resources (such as Wikipedia) can be deployed for NED. Based on the context of identified entities and information retrieved from Wikipedia we disambiguate entities and show which parameter configurations provide the best accuracy.

The paper is organized as follows. The next section describes the current state of the art and presents the most popular methods for NED. Subsequently, we introduce our approach. Based on the context of identified entities and data retrieved from Wikipedia we use text similarity measures to perform NED. The approach is analyzed on a collection of German news articles and on the Kulkarni name corpus[1]. The evaluation shows that our approach provides high quality results. Finally we draw a conclusion and give an outlook to future work.

2 Fundamentals

The concept of named entities (NE) was first introduced by the Message Understanding Conferences (MUC) evaluations [Sundheim1996]. The entities were limited to numerical expressions and a few classes of names. However, along with the development of information extraction and search technologies, the categories used for NEs were extended. Nowadays, complex ontologies (such as the Suggested Upper Merged Ontology, containing about 1,100 most general concepts [Pease and Niles2002]) are considered as the basis for named entities.

Current strategies for named entity recognition and disambiguation are mostly based on the use of ontological data and on context analysis. Since corpus based approaches are likely to suffer from the data sparseness problem, large lexical data collections, such as Wikipedia are a good choice to mine concepts and to enrich sparse corpora. Wikipedia is the largest online encyclopedia which provides linked and partially annotated data and descriptive information. Based on the evaluation of articles types (e.g. disambiguation pages, category articles) and the analysis of templates (e.g. info boxes) even semantic knowledge can be extracted. A popular project aiming to provide structured information from Wikipedia is DBpedia [Lehmann *et al.*2009]. Some authors [Cui *et al.*2009] built domain specific taxonomies from Wikipedia by analyzing the URLs and internal links

[1] http://www.d.umn.edu/~pederse/namedata.html

in Wikipedia pages, such as category labels or info boxes. For extracting new facts from Wikipedia Wu & Weld [Wu and Weld2007] suggest a cascade of conditional random field models.

Beside the approaches based on structured, ontological knowledge, there are projects that use unstructured Wikipedia data for NED. Cucerzan [Cucerzan2007] retrieves for words ("surface forms") identified to represent relevant entities all potentially matching Wikipedia articles. The Wikipedia contexts that occur in the document and the category tags are aggregated into a string vector, which is subsequently compared with the Wikipedia entity vector (of categories and contexts) of each possible entity. Then the assignment of entities to surface forms is chosen that maximizes the similarity between the document vector and the entity vectors. Alternative approaches for named entity disambiguation are based on the co-occurring analysis of named entities [Nguyen and Cao2008], the analysis of word based features (e.g. part-of speech, pattern) [Mann and Yarowsky2003] or document meta-information [Nadeau and Sekine2007, Bunescu and Pasca2006].

3 Approach

In our approach we focus on multi-lingual data retrieved from Wikipedia. Our intention is to have a robust approach providing a constantly high disambiguation precision. Thus, we do not rely on semantic annotations in Wikipedia (e.g. info boxes) or DBpedia content due to the fact that this data is not available for all entities.

The developed NED component is part of a project[2] for the semantic clustering of news, implemented using the UIMA framework[3]. The identification of words representing potentially relevant entities is done using DBpedia data. The component searches for surnames in news articles (present in the DBpedia person dataset). The found surnames and the assigned DBpedia entities are used as input data for the disambiguation task.

For performing NED, we analyze various methods to retrieve context data for the potentially matching entities. Due to the fact that we perform the named entity recognition based on DBpedia we focus on Wikipedia as data source. We analyze the following four content extraction strategies:

- We extract the first section of the Wikipedia article and remove the stop words. Usually the first section contains the most important information of the respective article.
- We extract the first section of the Wikipedia article and remove the stop words. The whole text is converted to lower case characters when calculating the string similarity.
- We extract all words with capital letters from the first section of the Wikipedia article. This is done to restrict the content to proper nouns (stop words are removed).
- We extract all words of the (complete) Wikipedia article that link to Wikipedia articles. The idea behind this method is, that the linked words contain the data suitable for the NED.

[2]http://www.dai-lab.de/competence_centers/irml/projekte/spiga/
[3]http://uima.apache.org/

We extract these data from the German as well as from the English Wikipedia. We use language-specific stop word lists and rule based stemmers.

In the next step we calculate the similarity between the retrieved Wikipedia content and the analyzed news article. The similarity calculation is done with the following algorithms:

1. **Jaccard-Similarity:** The Jaccard Similarity calculates the similarity of two word vectors (X, Y) as follows:
$$\text{Jaccard}(X,Y) = \frac{X * Y}{|X|\,|Y| - (X * Y)}$$
where $(X * Y)$ is the inner product of X and Y, and $|X| = \sqrt{X * X}$ the Euclidean norm of X.
2. **Dice-Similarity:** The Dice coefficient is a term based similarity measure whereby the similarity measure is defined as twice the number of terms common to compared entities divided by the total number of terms in both tested entities. The coefficient result of 1 indicates identical vectors as where a 0 equals orthogonal vectors.
$$\text{Dices coefficient}(X,Y) = \frac{2 * \#\text{common terms}}{\#\text{terms(X)} + \#\text{terms(Y)}}$$
3. **Overlap-Coefficient:** The Overlap-Coefficient calculates the similarity based on the common terms:
$$\text{Overlap}(X,Y) = \frac{\#\text{common terms}(X,Y)}{min\left(\#\text{terms}(X), \#\text{terms}(Y)\right)}$$
4. **Weighted Term-Similarity**: Similarity based on the number of matched terms weighted by the term length and the percentage of matched terms.
$$\text{wTSim}(X,Y) = \sqrt{|T|} * |X| * \sum_{t \in T} \left(\#(t \text{in} Y) \left(1 + \log\left(\frac{1}{1 + \#t\,(\text{in}Y)} \right) \right)^2 \text{len}(t) \right)$$
where T is the set of common terms in X and Y and $\text{len}(t)$ the function that calculates the length of term t. The weighted term similarity is often used by search engines for calculating a relevance score.

We calculate for each entity the similarity between the retrieved content and the news article. For the disambiguation task we determine the entity with the highest similarity. Overall, we test 32 different variants (4 content extraction strategies $\times$ 4 similarity measures $\times$ 2 languages). Additionally, we create ensembles combining the results of different languages (based on the CombSUM algorithm [Hull *et al.*1996]).

4 Experiments and evaluation

We evaluate the performance of our approach on a corpus of German news documents as well as on a collection of web search results in English.

4.1 Evaluation on the Kulkarni name corpus

The Kulkarni name corpus was created to evaluate NED algorithms. The corpus contains a set of queries where each query consists of a query string, and a set of approximately 200 documents (retrieved from a search engine for the query string). The query strings represent ambiguous person names. The documents in the result set are manually

annotated with the DBpedia URL of the entity related to the document. We performed the NED as described in section 3 and calculate the accuracy[4] over all queries. We consider only the English Wikipedia as content source since the relevant entities in the Kulkarni name corpus are not present in the German Wikipedia.

The accuracy for the analyzed content extraction strategies and the considered similarity measures are shown in Figure 1.

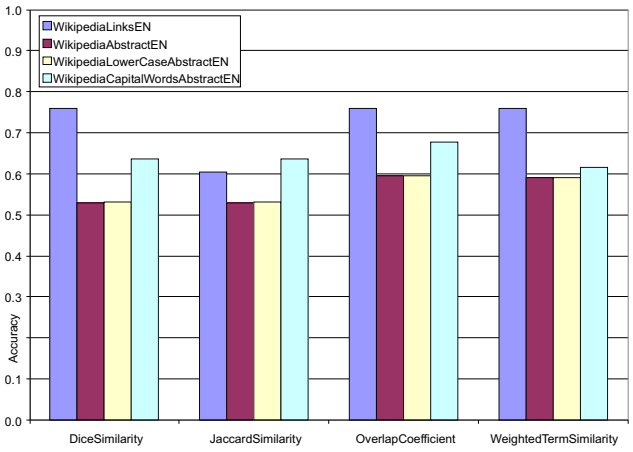


Figure 1. The named entity disambiguation accuracy on the Kulkarni name corpus

The results show that the disambiguation accuracy highly depends on the retrieved Wikipedia content. The best results are obtained if the disambiguation is based on the linked words in the Wikipedia article. The applied similarity measure has only a small influence on the result. Overall, the overlap coefficient and the weighted term similarity provide the best result. Thus, for a further improvement of the disambiguation accuracy the content extraction strategies should be optimized e.g. by considering the different types of Wikipedia links.

4.2 Evaluation on German news articles

Due to the fact, that the focus of our project is to cluster German news documents semantically, we create a new corpus optimized on our scenario. We randomly selected 65 news articles (from January 2009 crawled from various internet news sources, e.g. `Netzzeitung` and `DiePresse.com`) covering the topics politics and sports. Based on the German DBpedia person corpus[5] we identified potentially relevant people (having a surname present in the news document). We manually annotated which of the identified entities are related to the news article. Thus, each corpus element consists of the following data:

- The news document as plain text (⌀390 words)
- The search string (surname of the person), e.g. `Schumacher`
- A list of DBpedia entities having the search string as surname (limited to 10 entities), e.g. `[Michael Schumacher, Brad Schumacher, Anton Schumacher, Heinrich Christian ...]`

For the disambiguation, we considered the German and the English Wikipedia. The disambiguation accuracy for the considered content extraction strategies and the similarity measures are shown in Figure 2.

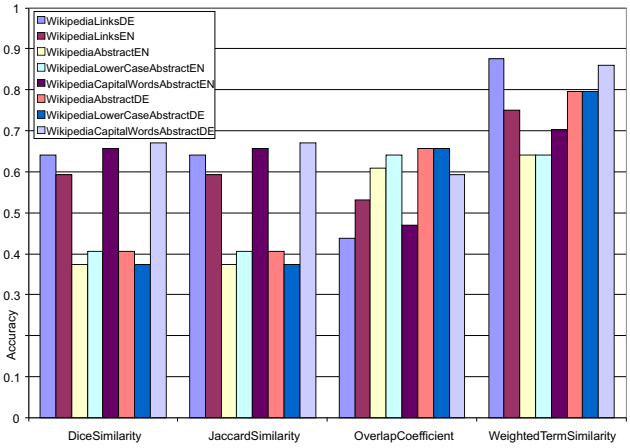


Figure 2. The named entity disambiguation accuracy on the German news corpus.

The evaluation shows that the highest accuracy is achieved when using the weighed term similarity and the linked words in the German Wikipedia page as reference content. In the analyzed scenario the similarity measure has a higher impact on the accuracy than the content extraction strategy. As expected, the accuracy achieved using the German Wikipedia content is always above the accuracy achieved based on the English content (since we analyzed German news). Nevertheless, the accuracy based on the analysis of the linked words in the English Wikipedia provided relatively good results. Comparing the dependencies between the accuracy from the content extraction strategies, the evaluation shows that good results are obtained if only relevant terms from the content source are filtered.

4.3 Multi-lingual ensembles

We analyze how the disambiguation accuracy can be improved by combining German and English content extraction strategies. Based on the strategy CombSUM we build ensembles combining the strategies discussed in section 4.2. The ensembles are created incrementally adding in each step the strategy to the ensemble that enables the best accuracy improvement. The weights for each strategy in the ensemble are calculated using an optimization algorithm based on genetic algorithms. The evaluation (Figure 3) shows that the weighted combination of strategies extracting data from the German and the English Wikipedia can improve the disambiguation accuracy by 13%. If English Wikipedia articles are used for the NED in German news articles the similarity is implicitly restricted to proper nouns (such as person or location names) that are not translated. Even though the strategies based only on the English Wikipedia content do not show a high disambiguation accuracy, in a combination with other strategies they improve the accuracy.

5 Conclusion and future work

We analyzed how multi-lingual content retrieved from Wikipedia can be used for NED. For evaluating the algorithms on German and English documents, we created a corpus of German news documents. Based on an English and a German corpus we analyzed various string similarity

[4] $\text{accuracy} = \frac{\text{number of correctly assigned entities}}{\text{total number of entities}}$

[5] http://downloads.dbpedia.org/3.5.1/de/persondata_de.nq.bz2

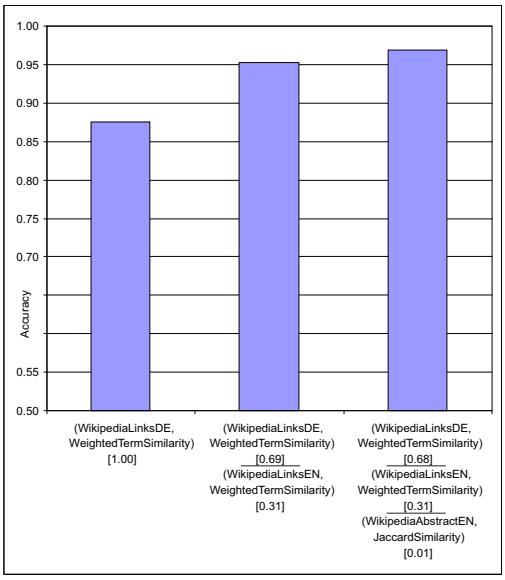


Figure 3. The disambiguation accuracy of the analyzed ensembles on the German news corpus.

measures and several strategies for content extraction. The evaluation results show, that in the chosen scenarios a high disambiguation accuracy can be obtained. The accuracy highly depends on the content extraction strategy. In general, the more complex extraction strategies provide better results.

Moreover, we showed that the combination of multilingual data improves the accuracy. The combination of German and English content extraction strategies improves the accuracy up to 13%. The combination of German and English content gives proper nouns a higher weight what results in a better disambiguation accuracy. A deeper analysis how the used parameter settings and the article language influence the NED accuracy will be done in the near future.

As future work we will take a deeper look on the relationships of the relevant named entities. The goal is to integrate semantic data sources (such as Freebase[6], or YAGO[7]) and to combine the data with multi-lingual Wikipedia data. We want to analyze the semantic relationships between the entities and learn weights for all relevant types of connections between entities. The learned weighs are used to adapt the similarity measures to the context and the respective domains. Moreover, in the project we will focus on ensemble learning algorithms [Polikar2006] (such as preference learning [Tsai *et al.*2007] and boosting strategies [Freund and Schapire1996]) to combine multi-lingual data and various features.

References

[Bunescu and Pasca, 2006] Razvan C. Bunescu and Marius Pasca. Using encyclopedic knowledge for named entity disambiguation. In *Proc. of the 11th Conference of the EACL)*. The Assn. for Computer Linguistics, 2006.

[Cucerzan, 2007] Silviu Cucerzan. Large-scale named entity disambiguation based on Wikipedia data. In *Proc. of the 2007 Joint Conf. on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 708–716, Prague, Czech Republic, June 2007. Assn. for Computational Linguistics.

[Cui *et al.*, 2009] Gaoying Cui, Qin Lu, Wenjie Li, and Yirong Chen. Mining concepts from wikipedia for ontology construction. In *Proc. of the 2009 Intl. Joint Conf. on Web Intelligence and Intelligent Agent Technology WI-IAT '09*, pages 287–290, Washington, DC, USA, 2009. IEEE Computer Society.

[Fellbaum, 1998] C. Fellbaum. *WordNet: An Electronical Lexical Database*. The MIT Press, Cambridge, MA, 1998.

[Freund and Schapire, 1996] Yoav Freund and Robert E. Schapire. Experiments with a new boosting algorithm. In *Intl. Conference on Machine Learning*, pages 148–156, 1996.

[Hull *et al.*, 1996] David A. Hull, Jan O. Pedersen, and Hinrich Schütze. Method combination for document filtering. In *SIGIR '96: Proc. of the 19th ACM SIGIR conf. on Research and development in information retrieval*, pages 279–287, New York, USA, 1996. ACM Press.

[Lehmann *et al.*, 2009] Jens Lehmann, Chris Bizer, Georgi Kobilarov, Sren Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. DBpedia - a crystallization point for the web of data. *Journal of Web Semantics*, 7(3):154–165, 2009.

[Mann and Yarowsky, 2003] Gideon S. Mann and David Yarowsky. Unsupervised personal name disambiguation. In *Proc. of the 7th conference on Natural language learning at HLT-NAACL 2003*, volume Volume 4, pages 33 – 40, Edmonton, Canada, 2003. Assn. for Computational Linguistics Morristown, NJ, USA.

[Nadeau and Sekine, 2007] David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Linguisticae Investigationes*, 30(1):3–26, January 2007. Publisher: John Benjamins Publishing Co.

[Nguyen and Cao, 2008] H.T. Nguyen and T.H. Cao. Named entity disambiguation on an ontology enriched by wikipedia. In *Research, Innovation and Vision for the Future*, pages 247 –254, 2008.

[Pease and Niles, 2002] Adam Pease and Ian Niles. Ieee standard upper ontology: a progress report. *Knowl. Eng. Rev.*, 17(1):pages 65–70, 2002.

[Polikar, 2006] Robi Polikar. Ensemble based systems in decision making. *Circuits and Systems Magazine, IEEE*, 6(3):21 – 45, 9 2006. 1531-636X.

[Sundheim, 1996] Beth M. Sundheim. The message understanding conferences. In *Proc. of the TIPSTER Text Program: Phase II*, pages 35–37, Vienna, VA, USA, May 1996. Assn. for Computational Linguistics.

[Tsai *et al.*, 2007] Ming-Feng Tsai, Tie-Yan Liu, Tao Qin, Hsin-Hsi Chen, and Wei-Ying Ma. FRank: a ranking method with fidelity loss. In *SIGIR '07: Proc. of the 30th ACM SIGIR conf. on Research and development in information retrieval*, pages 383–390, New York, USA, 2007. ACM.

[Wu and Weld, 2007] Fei Wu and Daniel S. Weld. Autonomously semantifying wikipedia. In *CIKM '07: Proc. of the 16th ACM conf. on information and knowledge management*, pages 41–50, New York, USA, 2007. ACM.

[6] http://www.freebase.com/

[7] http://www.mpi-inf.mpg.de/yago-naga/yago/

Implications of Inter-Rater Agreement on a Student Information Retrieval Evaluation

Philipp Schaer, Philipp Mayr, Peter Mutschke
GESIS – Leibniz Institute for the Social Sciences
53113, Bonn, Germany
philipp.schaer@gesis.org

Abstract

This paper is about an information retrieval evaluation on three different retrieval-supporting services. All three services were designed to compensate typical problems that arise in metadata-driven Digital Libraries, which are not adequately handled by a simple *tf-idf* based retrieval. The services are: (1) a co-word analysis based query expansion mechanism and re-ranking via (2) Bradfordizing and (3) author centrality. The services are evaluated with relevance assessments conducted by 73 information science students. Since the students are neither information professionals nor domain experts the question of inter-rater agreement is taken into consideration. Two important implications emerge: (1) the inter-rater agreement rates were mainly fair to moderate and (2) after a data-cleaning step which erased the assessments with poor agreement rates the evaluation data shows that the three retrieval services returned disjoint but still relevant result sets.

1. Introduction

Metadata-driven Digital Libraries (DL) face three typical difficulties: (1) the vagueness between search and indexing terms, (2) the information overload by the amount of result records returned by information retrieval systems, and (3) the problem that pure term frequency based rankings, such as term frequency – inverse document frequency (*tf-idf*), provide results that often do not meet user needs [Mayr *et al.*, 2008]. To overcome these limitations the DFG-funded project "Value-Added Services for Information Retrieval" (IRM[1]) is doing research on an overall approach to use computational science models as enhanced search stratagems [Bates, 1990] within a scholarly retrieval environment, which might be implemented within scholarly information portals.

To show that a user's search improves by using these model-driven search services when interacting with a scientific information system, an information retrieval evaluation was conducted. Since the assessors in this experiment were neither information professionals nor domain experts we had especially looked at the level of agreement between the different assessors. This also is our special interest since in big evaluation campaigns like CLEF only a minority of the evaluations made their evaluations with regard to the inter-rater agreement. In the CLEF campaign of 2009 no analysis of inter-rater agreement was made in the classical ad-hoc track [Ferro and Carol, 2009] and only one to four assessors had to judge the pooled documents. The only track that made these inter-rater analyses was the medical imaging track [Müller *et al.*, 2009], but they only had two assessors per topic. These numbers are quite low compared to our evaluation where for some topics there were up to 15 assessors. With the high number and the non-professional background of our assessors a statistical measure is needed to rate the reliability and consistency of agreement between the different assessments.

Several metrics have been developed to measure inter-grader reliability, like mean overlap or Fleiss' Kappa, which will be briefly presented in the next section. After that we will introduce the three services that are based on the principles of co-words, core journals and centrality of authors. The basic assumptions and concepts are presented. The conducted evaluation and study with 73 participants is described in the following section. The paper closes with a discussion of the observed results.

2. Inter-rater Agreement

The reliability and consistency of agreement between different assessments can be measure by a range of statistical metrics. Two of these metrics to measure inter-grader reliability are for example mean overlap or Fleiss' Kappa.

2.1. Mean Overlap and Overall Agreement

In the early TREC conferences (namely till TREC 4) a rather simple method was used to measure the amount of agreement among different assessors by calculating the overlap between the different assessors judgements. Overlap was defined by the size of the intersection divided by the size of the union of the relevant document sets. Despite the relatively high average overlap, Voorhees [2000] reported that there were significant different judgements for some topics. She reported that in TREC topic 219 for example there was no single intersection between two assessors at all. Therefore the mean overlap in this TREC topic was between 0.421 and 0.494 comparing two assessors directly and 0.301 comparing the set of all assessors together.

This measure was discarded after TREC 4. One of the reasons was that this measure gets very unstable when more than three assessors are taken into account.

[1] http://www.gesis.org/irm

2.2. Fleiss' Kappa

Fleiss' Kappa is a measure of inter-grader reliability – based on Cohen's Kappa – for nominal or binary ratings [Fleiss, 1971]. The Kappa value can be interpreted as the extent to which the observed amount of agreement among raters exceeds what would be expected if all raters made their ratings completely randomly.

Fleiss' Kappa is given in equations 1 to 3:

$$\kappa = \frac{\overline{P} - \overline{P}_e}{1 - \overline{P}_e} \quad (1)$$

where

$$\overline{P} = \frac{1}{Nn(n-1)} \sum_{i=1}^{N} \sum_{j=1}^{k} n_{ij}(n_{ij} - 1) \quad (2)$$

$$\overline{P}_e = \sum_{j=1}^{k} \left(\frac{1}{Nn} \sum_{i=1}^{N} n_{ij} \right)^2 \quad (3)$$

N is the total number of subjects (e.g. documents to be assessed); n is the number of judgments per subject (raters); k is the number of response categories.

Kappa scores can range from 0 (no agreement) to 1.0 (full agreement). Landis and Koch [1977] suggest interpreting the score as followed: $\kappa<0$ = poor agreement, $0\leq\kappa<0.2$ = slight agreement, $0.2\leq\kappa<0.4$ = fair agreement, $0.4\leq\kappa<0.6$ = moderate agreement, $0.6\leq\kappa<0.8$ = substantial agreement, $0.8\leq\kappa\leq1$ = (almost) perfect agreement. These interpretations are not generally accepted and other interpretations are possible. Greve and Wentura [1997] suggest to interpret scores $\kappa<0.4$ as "not be taken too seriously" and values of $0.4\leq\kappa<0.6$ as acceptable. $0.75\leq\kappa$ seems good up to excellent.

3. Evaluated System and Services

All proposed models are implemented in a live information system using (1) the Solr search engine, (2) Grails Web framework and (3) Recommind Mindserver to demonstrate the general feasibility of the approaches. Solr is an open source search platform from the Apache Lucene project[2], which uses a *tf-idf*-based ranking mechanism[3]. The Mindserver is a commercial text categorization tool, which was used to generate the query expansion terms. Both Bradfordizing and author centrality as re-rank mechanism are implemented as plugins to the open source web framework Grails, which is the glue to combine the different modules and to offer an interactive web-based prototype[4].

3.1. Query Expansion by Search Term Recommendation

When using search in an information system a user has to come up with the "correct" terms to formulate his query. These terms have to match the terms used in the documents or in the description of the documents to get an appropriate result. Especially in the domain of metadata-driven Digital Libraries this is long known as the language problem in IR [Petras 2006].

Figure 1: Screenshot of the web-based assessment tool. The users were shown the name and description of the task (top) and a number of documents to assess. The documents (middle) had author, publication year, title, abstract and keywords, which the assessors could use to judge the documents relevant or non-relevant (right).

The Search Term Recommender (STR) is based on statistical co-word analysis and builds associations between free terms (i.e. from title or abstract) and controlled terms (i.e. from a thesaurus). Controlled terms are assigned to the document during a professional indexation and enrich the available metadata on the document. The co-word analysis implies a semantic association between the free and the controlled terms. The more often terms co-occur in the text the more likely it is that they share a semantic relation. The commercial classification software Recommind Mindserver, which is based on Support Vector Machines (SVM) and Probabilistic Latent Semantic Analysis (PLSA), calculated these associations. The software was trained with the SOLIS database to best match the specialized vocabularies used in the Social Sciences.

When used as an automatic query expansion service the Mindserver's top n=4 suggested terms were taken to enhance the query. The query was expanded by simply OR-ing them. If we take a sample query on Poverty in Germany, it is expanded to:

```
povert* AND german* →

(povert* AND german*) OR "poverty" OR
"Federal Republic of Germany" OR "so-
cial assistance" OR "immiseration"
```

3.2. Re-Ranking by Bradfordizing

The Bradfordizing re-ranking service addresses the problem of oversized result sets by using a bibliometric method. Bradfordizing re-ranks a result set of journal articles according to the frequency of journals in the result set such that articles of core journals – journals which publish frequently on a topic – are ranked higher than articles from other journals. This way the central publication sources for any query are sorted to the top positions of the result set [Mayr, 2009].

[2] http://lucene.apache.org/Solr/
[3] http://lucene.apache.org/java/2_4_0/scoring.html
[4] http://www.gesis.org/beta/prototypen/irm/

			Fleiss' Kappa				Mean overlap	
topic	title	description	*n*	*N*	*k*	κ	>=0.8	=1
83	Media and War	Find documents on the commentatorship of the press and other media from war regions.	15	40	2	0.522	0.727	0.225
84	New Media in Education	Find documents reporting on benefits and risks of using new technology such as computers or the Internet in schools.	11	40	2	0.304	0.5	0.15
88	Sports in Nazi Germany	Find documents about the role of sports in the German Third Reich.	6	40	2	0.528	0.75	0.425
93	Burnout Syndrome	Find documents reporting on the burnout syndrome.	10	40	2	0.411	0.8	0.35
96	Costs of Vocational Education	Find documents reporting on the costs and benefits of vocational education.	2	40	2	0.488	0.775	0.775
105	Graduates and Labour Market	Find documents reporting on the job market for university graduates.	5	40	2	0.466	0.675	0.525
110	Suicide of Young People	Find documents investigating suicides in teenagers and young adults.	5	40	2	0.222	0.625	0.425
153	Childlessness in Germany	Information on the factors for childlessness in Germany	10	40	2	0.202	0.325	0.175
166	Poverty in Germany	Research papers and publications on poverty and homelessness in Germany.	9	40	2	0.438	0.5	0.25
173	Propensity towards violence among youths	Find reports, cases, empirical studies and analyses on the capacity of adolescents for violence.	10	40	2	0.411	0.55	0.2
				avg.		*0.4*	*0.622*	*0.35*

Table 1: Ten CLEF topics and the corresponding inter-grader reliability expressed by Fleiss' Kappa and mean overlap per topic. The mean overlap is calculated using two different thresholds (>=0.8 means that an intersection rate of 80% is counted and =1 means that only perfect matches are counted).

In a first step the search results are filtered with their ISSN, since ISSNs are proper identifiers for journals. The next step aggregates all results with the same ISSN. For this step the build-in faceting mechanism of Solr is used. The journal with the highest ISSN facet count gets the top position in the results; the second journal gets the next position, and so on. In the last step, each documents rank (given through Solr's internal ranking) is boosted by the frequency counts of the journals. This way all documents from the journal with the highest ISSN facet count are sorted to the top.

3.3. Re-Ranking by Author Centrality

Author centrality is another way of re-ranking result sets. Here the concept of centrality of authors in a network is an approach for the problem of large and unstructured result sets. The intention behind this ranking model is to make use of knowledge about the interaction and cooperation behavior in fields of research. The model assumes that the relevance of a publication increases with the centrality of their authors in co-authorship networks. The user is provided with publications of central authors when the result set obtained is re-ranked by author centrality.

The re-ranking service calculates a co-authorship network based on the result set to a specific query. Centrality of each single author in this network is calculated by applying the betweenness measure and the documents in the result set are ranked according to the betweenness of their authors so that publications with very central authors are ranked higher in the result list. Since the model is based on a network analytical view it differs greatly from text-oriented ranking methods like *tf-idf* [Mutschke, 2004].

4. Evaluation

We conducted a user assessment with 73 information science students who used the SOLIS database[5] with 369,397 single documents on Social Science topics to evaluate the performance of the three presented services. The documents include title, abstract, controlled keywords etc. The assessment system, which was built on top of the IRM prototype described earlier and all documents were in German. All written examples in this paper are translated.

4.1. Method

A standard approach to evaluate Information Retrieval systems is to do relevance assessments, where – in respect to a defined information need – documents are marked as relevant or not relevant. There are small test collections (like Cranfield) where all containing documents are judged relevant or not relevant. For large collections (like TREC, CLEF etc.) this is not possible, so only subsets are assessed. Only the top n documents returned by the different retrieval systems are assessed. Pooling is used to disguise the origin of the document [Voorhees and Harman, 2005].

In our assessment the participants were given a concrete search task, which was taken from the CLEF corpus. After a briefing each student had to choose one out of ten different predefined topics (namely CLEF topics 83, 84, 88, 93, 96, 105, 110, 153, 166 and 173). Topic title and the description were presented to form the information need (cp. table 1 and figure 1).

[5] SOLIS is accessable via the Sowiport portal http://www.gesis.org/sowiport

The pool was formed out of the top n=10 ranked documents from each service and the initial *tf-idf* ranked result set respectively. Duplicates were removed, so that the size of the sample pools was between 34 and 39 documents each. The assessors could choose to judge relevant or not relevant (binary decision) – in case they didn't assess a document this document was ignored in later calculations.

4.2. Data Set

The 73 assessors, who were information science students, did 43.78 assessments in average. They did 3,196 single relevance judgments in total. Only 5 participants didn't fill out the assessment form completely. Since every assessor could freely choose from the topics the assessments are not distributed evenly. Topic 83 was picked 15 times – topic 96 twice. It was a conscious decision to allow each assessor to freely choose a topic he or she is familiar with, but this had direct consequences on the number of single assessments (cp. table 2).

5. Results

Since the assessors in this experiment were neither information professionals nor domain experts the results should be discussed with a special interest on the level of agreement between the assessors. To rate the reliability and consistency of agreement between the different assessments we used mean overlap and Fleiss' Kappa, described in section 2. Besides that precision of each topic/ service combination and the intersection of the different result sets was calculated.

5.1. Mean Overlap and Overall Agreement

Normally only a true overlap is counted. When comparing the judgements of two assessors they can either agree or disagree. When dealing with more than two assessors the situation gets difficult: What to do if 14 assessors agree but one disagrees? To take majority decision into account we applied two different thresholds calculating the mean overlap: 1 and 0.8. Only perfect matches count when a threshold of 1 is applied: This means that all assessors have to agree 100%. Here the measured mean overlap (140 intersections where all assessors agreed 100% in a total union set of 400) was 0.35 (cp. table 1). This value can be compared to the reported values from the selected TREC topic (see section 2.1) where the overlap was between 0.421 and 0.494 comparing two assessors directly and 0.301 comparing the set of all three assessors together.

The more assessors per topic the lower the mean overlap gets. This is quite natural since 100% agreement is easier to obtain between 2 assessors (for example for topic 96) than for 15 assessors (like for example in topic 83). Therefore a second threshold of 0.8 was applied. Here all judgements with only slight disagreement where also taken into consideration. This had direct implications on the average overlap value over all assessments. The average overlap rate was 0.622.

5.2. Fleiss' Kappa

The equation for Fleiss' Kappa is given in section 2.2. In our study N was always forty (ten documents for each value added service – here assessments for duplicates count for all services that returned it) and k was always two (binary decision).

All Kappa scores in our experiment range between 0.20 and 0.52, which are fair up to moderate levels of agreement or mainly acceptable in the more conservative interpretation (cp. table 1). The average Kappa value was 0.4.

5.3. Precision

The precision *P* was calculated by equation 4:

$$P = \frac{|\{rel\} \cap \{ret\}|}{|\{ret\}||} \quad (4)$$

P was calculated for each topic and service, where $|\{rel\}|$ is the number of all relevant assessed documents and $|\{ret\}|$ is the number of all retrieved documents which were in the pool (relevant and not relevant). All unfiltered precision values and numbers of relevance assessments can be seen in table 2 and figure 2.

The average precision of the STR was highest (68%) compared to the baseline from the SOLR system (57%). The two alternative ranking methods Bradfordizing (BRAD) and author centrality (AUTH) scored 57% and 60% respectively.

We calculated additional average precision values and left out the topics marked unstable by the mean overlap values (those with a value of <=0.35 in case of a 100% match). Here the topics 83, 84, 153, 166 and 173 were left out. This was not considered useful since only half the

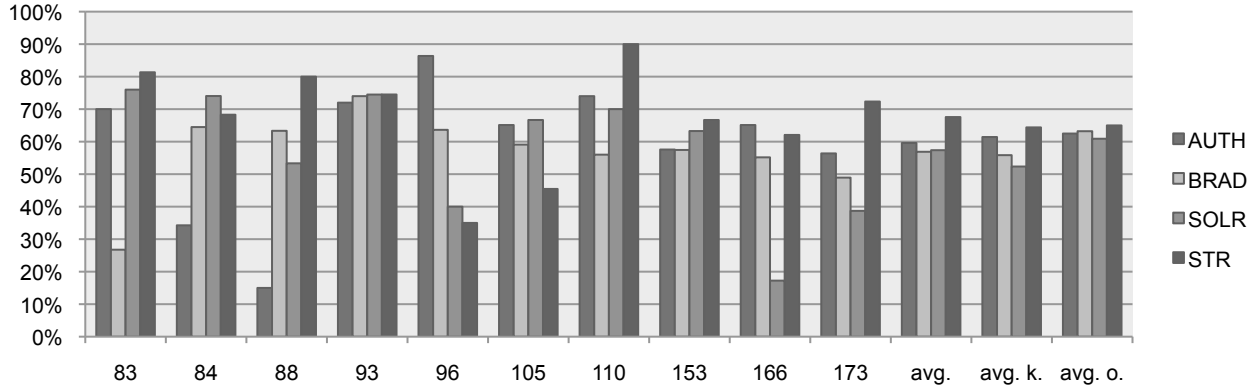


Figure 2: Precision for each topic and service (Relevance assessments per topic / total amount of single assessments). The three average precision values are (1) avg.: unfiltered values, (2) avg. k.: average computed with a kappa threshold of 0.4 and (3) avg. o.: average computed with a overlap threshold of 0.35.

data set remains after this data clearance step. Nevertheless calculating the average precision for each service without the five topics would have been: AUTH: 63%, BRAD: 63%, SOLR 61% and STR: 65%.

Considering the Kappa values the relevance judgments from topics 84, 110 and 153 were below the threshold of 0.40 and so they might have been dropped. Calculating the average precision for each service without the three unstable topics would have been: AUTH: 61%, BRAD: 56%, SOLR 52% and STR: 64% (cp. table 2).

5.4. Intersection of result sets

A comparison of the intersection of the relevant top 10 document result sets between each pair of retrieval service shows that the result sets are nearly disjoint. 400 documents (4 services * 10 per service * 10 topics) only had 36 intersections in total (comp. figure 3). Thus, there is no or very little overlap between the sets of relevant top documents obtained from different rankings.

AUTH and SOLR as well as AUTH and BRAD have just three relevant documents in common (for all 10 topics), and AUTH and STR have only five documents in common. BRAD and SOLR have six, and BRAD and STR have five relevant documents in common. The largest, but still low overlap is between SOLR and STR, which have 14 common documents.

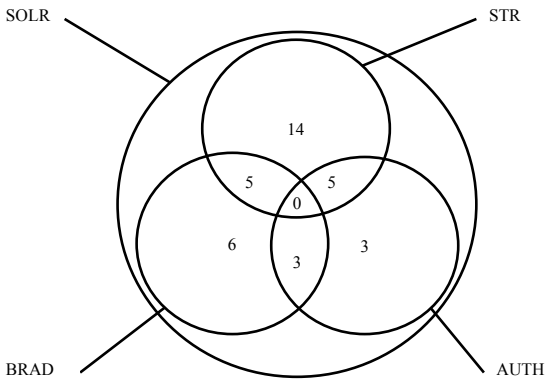


Figure 3: Intersection of top n=10 documents from all topics and services (total of 400 documents). The number 3 in the lower right circle for example means that only 3 documents in the result set returned by the SOLR system and the result set returned by the author centrality service AUTH are the same.

6. Discussion

The discussion will focus on two different aspects: (1) the results of the inter-rater agreement and our implications to our evaluation setup and (2) the evaluation itself and the outcomes regarding precision and intersection of result sets.

6.1. Inter-rater agreement

The central question we have to consider is to what amount the evaluation is significant before we can interpret the results in form of precision values. At first our results looked promising, when comparing them to the mean overlap agreement rate from the TREC studies, especially when taking the high number of assessors per topics into account. After we had a look at the mean overlap and Kappa values we had to differentiate more on the single topics since their performance was not simply comparable.

Taking the non-professional assessors and the Kappa values in consideration we have to ask ourselves how reliable our evaluations are. The general setup is not unusual: Other studies also relied on non-professional assessors and showed promising results: Al-Maskari *et al.* [2008] observed a rather high overall agreement between official TREC and non-TREC assessors in a study with 56 participants and 4399 single document assessments in total. The agreement changed due to the different topics and the actual ranking position of the assessed document and was between 75 and 82% if both relevant and irrelevant documents assessments were counted. Alonso and Mizzaro [2009] compared official TREC assessments with anonymous assessors from Amazon Mechanical Turk. They observed a rather small sample of 10 participants and 290 judgments. For the relevant documents the average across of participants was 91% and in case of not relevant documents the average was 49%.

	non relevant				relevant				precision			
topic	AUTH	BRAD	SOLR	STR	AUTH	BRAD	SOLR	STR	AUTH	BRAD	SOLR	STR
83	42	104	36	25	98	38	114	109	0.70	0.27	0.76	0.81
84	71	38	27	26	37	69	77	56	0.34	0.64	0.74	0.68
88	51	22	28	12	9	38	32	48	0.15	0.63	0.53	0.80
93	28	26	25	26	72	74	73	76	0.72	0.74	0.74	0.75
96	3	8	12	13	19	14	8	7	0.86	0.64	0.40	0.35
105	15	18	15	24	28	26	30	20	0.65	0.59	0.67	0.45
110	13	22	15	5	37	28	35	45	0.74	0.56	0.70	0.90
153	42	40	36	32	57	54	62	64	0.58	0.57	0.63	0.67
166	30	39	72	33	56	48	15	54	0.65	0.55	0.17	0.62
173	41	48	57	26	53	46	36	68	0.56	0.49	0.39	0.72
								avg.	*0.60*	*0.57*	*0.57*	*0.68*
								avg.(kappa >= 0.4)	*0.61*	*0.56*	*0.52*	*0.64*
								avg. (overlap >= 0.35)	*0.63*	*0.63*	*0.61*	*0.65*

Table 2: Relevance judgments for each topic and service (total number of non-relevant and relevant judgments) and the calculated precision values (with different thresholds applied).

Nevertheless a mandatory step should be to sort out subsets of the assessment data where Kappa values (or other reliable measures) are below a certain threshold. Unfortunately this is not done in all studies.

6.2. Precision and Intersection

Comparing the precision values of the different approaches the three services in average all performed better (in case of the filtered topic list) or at least same (when counting all assessments) as the naïve *tf-idf* baseline. This is true for all average precision values calculated regarding all kinds of thresholds and data clearing.

When inspected on a per-topic basis the performance is more diverse. While the STR outperforms in nearly all cases the other services had topics where their precision was significantly smaller compared to the other sources. For topic 83 the Bradfordizing re-ranking service couldn't sort the result set appropriately while the author centrality service couldn't adequately handle topic 88.

Discussing the results of the two proposed re-ranking methods Bradfordizing and author centrality brings up two central insights: (1) users get new result cutouts with other relevant documents which are not listed in the first section (first n=10 documents) of the original list and (2) Bradfordizing and author centrality can be a helpful information service to positively influence the search process, especially for searchers who are new on a research topic and don't know the main publication sources or the central actors in a research field. The STR showed an expected behavior: While the result set in total increases the first n=10 hits are more precise. This is quite normal in query expansion scenarios where the high descriptive power of the controlled terms that are added to the query increases the precision [Efthimiadis, 1996]. This is an indicator for the high quality of the semantic mapping between the language of scientific discourse (free text in title and abstract) and the language of documentation (controlled thesauri terms).

The very low overlap of the result sets as described in section 5.4 confirms that the value-added services proposed provide a quite different view to the document space: Not only from a term-centric view proposed by *tf-idf* (with or without query expansion mechanism) but also from a more person- or journal-centric perspective. Which is even more interesting since the average precision values didn't differ as much as one might have expected.

7. Outlook

After the evaluation and the analysis regarding inter-rater agreement two important implications emerge: (1) the inter-rater agreement rates were mainly fair to moderate and therefore showed a general feasibility of a non-experts evaluation and (2) after a data-cleaning step which erased the assessments with very poor agreement rates the evaluation data showed that the three retrieval services returned disjoint but still relevant result sets. The services provide a particular view to the information space that is quite different from traditional retrieval methods.

Although the results looked promising our next step is a new relevance assessment with scientist and domain experts to have a direct comparison and to reinsure our observations.

Acknowledgments

Vivien Petras and Philipp Mayr supervised the 73 students in their courses at Humboldt University and University of Applied Science Darmstadt. Hasan Bas implemented the assessment tool during his internship with GESIS.

This work was funded by DFG (grant number INST 658/6-1).

References

[Alonso and Mizarro, 2009] Omar Alonso and Stefano Mizzaro. Can we get rid of TREC assessors? Using Mechanical Turk for relevance assessment. *Proceedings of the SIGIR 2009 Workshop on The Future of IR Evaluation* : 15-16, 2009.

[Al-Maskari *et al.*, 2008] Azzah Al-Maskari, Mark Sanderson, and Paul Clough. Relevance judgments between TREC and Non-TREC assessors. *Proceedings of the 31st annual international ACM SIGIR*, 2008.

[Bates, 1990] Marcia J. Bates. Where Should the Person Stop and the Information Search Interface Start? *Information Processing & Management*, 26 575-591, 1990.

[Efthimiadis, 1996] Efthimis N. Efthimiadis. *Query expansion. Annual Review of Information Systems and Technology (ARIST)*, Vol. 31, 121-187, 1996.

[Ferro and Carol, 2009] Nicola Ferro and Carol Peters. CLEF 2009 Ad Hoc Track Overview: TEL & Persian Tasks. *Working Notes for the CLEF 2009 Workshop, Corfu, Greece*, 2009.

[Fleiss, 1971] Joseph L. Fleiss. Measuring nominal scale agreement among many raters. *Psychological Bulletin, 76*(5), 378–382, 1971.

[Greve and Wentura, 1997] Werner Greve and Dirk Wentura. Wissenschaftliche Beobachtung: Eine Einführung. Weinheim, PVU/Beltz, 1997.

[Landis and Koch, 1977] J. Richard Landis and Gary G. Koch. The Measurement of Observer Agreement for Categorical Data. *Biometrics, 33*(1), 159-174, 1977.

[Mayr et al., 2008] Philipp Mayr, Peter Mutschke, and Vivien Petras. Reducing semantic complexity in distributed digital libraries: Treatment of term vagueness and document re-ranking. *Library Review*, 57(3), 213-224, 2008.

[Mayr, 2009] Philipp Mayr. Re-Ranking auf Basis von Bradfordizing für die verteilte Suche in Digitalen Bibliotheken. Humboldt-Universität zu Berlin, Berlin, 2009.

[Mutschke, 2004] Peter Mutschke. Autorennetzwerke: Netzwerkanalyse als Mehrwertdienst für Informationssysteme. In: *Information zwischen Kultur und Marktwirtschaft (ISI 2004)*, 141-162, 2004.

[Müller et al., 2009] Henning Müller, Jayashree Kalpathy–Cramer, Ivan Eggel, Steven Bedrick, Saïd Radhouani, Brian Bakke Jr., Charles E. Kahn, and William Hersh. Overview of the CLEF 2009 Medical Image Retrieval Track. *Working Notes for the CLEF 2009 Workshop, Corfu, Greece*, 2009.

[Petras, 2006] Vivien Petras. Translating Dialects in Search: Mapping between Specialized Languages of Discourse and Documentary Languages. University of California, Berkley, 2006.

[Voorhees, 2000] Ellen M. Voorhees. Variations in relevance judgments and the measurement of retrieval effectiveness. *Information Processing & Management*. 36(5): 697-716, 2000.

[Voorhees and Harman, 2005] Ellen M. Voorhees and Donna K. Harman (Eds.). *TREC: Experiment and Evaluation in Information Retrieval*: The MIT Press, 2005.

Evaluation of five web search engines in Arabic language

Wissam Tawileh, Thomas Mandl and Joachim Griesbaum
University of Hildesheim
D-31141, Hildesheim, Germany
tawilehw@uni-hildesheim.de

Abstract

To explore how Arab Internet users can find the information in their mother tongue on the web, the five web search engines Araby, Ayna, Google, MSN and Yahoo were tested on an information retrieval evaluation basis with the consideration of the web-specific evaluation requirements. The test used fifty randomly selected queries from the top searches on the Arabic search engine Araby. The relevance of the top ten results and their descriptions retrieved by each search engine for each query were evaluated by independent jurors. Evaluations of results and descriptions were then compared to assess their conformity. The core finding was that Google performed almost all the times better than the other engines. The difference to Yahoo was however not statically significant, and the difference to the third ranked engine MSN was significant to a low degree. The Arabic search engine Araby showed performance on most of the evaluation measures, while Ayna was far behind all other search engines. The other finding was the big differences between search results and their descriptions for all tested engines.

1 Introduction

This work represents an evaluation test performed on multiple web search engines which can deal with Arabic and the specific needs of this language and its speakers as the target users group considered in the test.

The question this test attempts to answer is: which web search engine of the five tested in this study can retrieve the "best" search results for the user judging these results? The test compares the effectiveness of five different web search engines, two of which are native Arabic engines: Araby and Ayna, and three are international Arabic-enabled engines Google, MSN and Yahoo.

The motivation of the test is the lack of evaluative research of Arabic information retrieval systems, especially on the internet. This is despite the very high growth rates of internet users in the Arab countries, and that most users can not read English which dominates the content on the World Wide Web [Hammo, 2009].

2 Related Work

The evaluation of web search engines has been gaining increased importance and research interest since its early beginnings in 1996. A large number of evaluation experiments has been performed to assess the performance of search engines from different perspectives using varied evaluation measures and test designs.

Most evaluation tests used search queries in English as a dominant language on the web. However, many tests in other languages are also taking place, focusing mainly on the way search engines deal with different languages, their linguistic issues and proper search algorithms aiming to improve the multilingual capabilities of search engines.

Other studies, like this work, focus on the performance evaluation of web search engines from the local users' point of view. An overview on non-English web information retrieval studies is presented by Lazarinis and others [Lazarinis, 2007; Lazarinis *et al.*, 2009].

Several studies (like [Moukdad and Large, 2001], [Moukdad, 2002; 2004], [Abdelali *et al.*, 2004] and recently [Hammo, 2009]) discussed Arabic information retrieval on the web.

Gordon and Pathak [1999] discussed a collection of web search engines evaluation tests conducted since 1996 against their test methodology and purpose.

Another overview aligned with the recommendations of Tague-Sutcliffe [1992] for general information retrieval evaluation, and Hawking et al. [2001] for web-specific information retrieval evaluation criteria is presented in [Lewandowski, 2008].

The most recent two studies similar to this work in their methodology and design are:

Griesbaum tested three search engines (Google, Lycos and AltaVista) using 50 queries, in German language [Griesbaum, 2004]. Google came in the first place in the overall performance judgment followed by Lycos with no significant difference, and AltaVista came in the last place with a higher difference to Google, but not to Lycos.

The second study has been conducted by [Lewandowski, 2008]. He evaluated five search engines (Google, Yahoo, MSN, Ask.com and Seekport). A set of forty queries, in German language, was created by forty faculty students who were the jurors as well. The study found no significant reason to favor any of the major search engines in terms of performance and concluded that more attention should be paid by search engine companies to the quality of results descriptions.

3 Test Methodology

Tague-Sutcliffe [1992] presented a guide for information retrieval evaluation which helps the experimenters in making the required decisions while planning an evaluation test to ensure the validity of the experiment, the reliability of the results and the efficiency of the test proce-

dures. It assumes to answer ten questions which are discussed here for this study.

3.1 Need for testing

Keeping in mind the limited knowledge of foreign languages, especially English, among Arab internet users, many questions can be stated considering the effectiveness and efficiency of searching the web in Arabic using local Arabic search engines or international Arabic enabled ones.
While these questions can cover the search systems themselves (search algorithms, language handling issues, resources consumption… etc.), another aspect can also be questioned, which is the informativeness and effectiveness of web search engines in the eyes of Arab internet users.
To the best knowledge of the authors, there is no similar published evaluation test before this work.

3.2 Type of test

The test evaluates five online commercial search engines in the Arabic language working on the web as an operational database.
This kind in real use differs from the experimental tests performed in a laboratory environment based on the Cranfield paradigm [Mandl, 2008]. Such tests allow a higher level of control on tests parameters and variables, whereas the test presented here gives an assessment closer to real life and the users' perception of results.

3.3 Variables definition

The independent variables which affect the results of this test are: evaluation criteria, relevance performance measures, queries, information needs and participants (also referred to as users or jurors). These variables are defined in this section to illustrate the general settings of the test.
The dependent variables are the relevance judgments of the jurors which are the indicator of the retrieval performance in this test. These are discussed in the results section.

Evaluation criteria

Like in most information retrieval tests, the main evaluation criterion of search results is relevance. This should measure the ability of tested web search engines to satisfy the users' information needs described in their search queries.
Relevance assessment is a problematic issue discussed in several studies, and can be influenced by many factors [Schamber, 1994]. A representative assessment can however be done by individual jurors to avoid bias of the researcher in the judgment process.
Search results are evaluated in this test on a binary basis to be "relevant" or "not relevant". Following [Lewandowski, 2008] a search result should satisfy the users' information needs without taking further actions.
Results descriptions are also evaluated on a binary basis as "seems relevant" or "seems not relevant".
To cover the possibility that a search engine may present a result without a description in its results list, the evaluation option "no description available" is also given to jurors.
As this test is specifically designed for Arabic search evaluation and targeting Arab internet users with information needs in their native language, all retrieved documents in languages other than Arabic should be evaluated as "not relevant". Jurors were instructed before they start to judge Non-Arabic results descriptions as "seems not relevant" as well.

Relevance performance measures

Precision is a standard information retrieval evaluation measure used in this test. The other standard evaluation measure, recall, used to evaluate the performance of classical information retrieval systems can not easily be applied to web search engines evaluations as the total number of relevant documents can not be estimated.
As most internet users usually look only at the first one or two results of a query from a search engine, a cut-off value of the first ten results can give reliable evaluation results using the so called top-ten precision.
Precision values will be calculated on both the macro and the micro levels [Womser-Hacker, 1989].

Queries

To be as close as possible to real-life search behavior of Arab web users, a random set of search queries is selected from the most used search queries on the Arabic web search engine (Araby.com).
A collection of fifty queries (a standard for TREC evaluation tests) is a reasonable amount for valid evaluation results. Additional ten queries were reserved for any problems that may occur during the test.
Queries were selected and executed exactly as typed by the original users (as listed in the search engine Araby on 10. March. 2009). No correction or alternative writing methods were suggested.

Information needs

After selecting the random set of search queries, a reconstruction of the information needs behind these queries is necessary to simulate the needs of users originally entered these queries in the search engine and form the relevance judgment criteria. This task is particularly difficult with general short queries.
A group of Arab internet users (mainly students and engineers) were asked to describe their needs of information when searching for given five different queries. All descriptions of each query were then merged to form the relevance judgment criteria. For the search query "Sayings" for example, relevant documents should contain: "Sayings of elders, politicians or celebrities".

Participants

Participants in this test had to be native Arabic speakers and to have average knowledge of internet browsing and usage of web search engines.
A total number of seventy volunteers (53 males and 17 females) filled out the information needs reconstruction forms. To avoid bias in the information needs simulation, users from multiple ages and different education background described information requirements they may associate with the given search queries.
The ideal number of evaluation jurors for fifty queries is equal to fifty, so that each juror can evaluate a single query on all tested search engines. Out of eighty nine invited users (friends and colleagues of the first author), the total number of fifty jurors (42 males and 8 females) from nine Arab countries was achieved.

3.4 Search engines selection

According to the recommendations in [Hawking *et al.*, 2001] for the evaluation of web search engines, the major

search engines should be included in the test. As this work tries to explore the suitability of native Arabic web search engines as alternatives to international market leading engines for Arab users, local search engines were tested in addition to the leading international engines.
The most popular five search engines in the Arab countries according to Alexa were selected, two search engines are native Arabic and three are international Arabic-enabled engines. The selected search engines are:

- Araby (www.araby.com)
- Ayna (www.ayna.com)
- Google (www.google.com.sa)
- MSN (www.live.com[1])
- Yahoo! (www.yahoo.com)

3.5 Finding queries

Although there is no published statistics about queries length and complexity for Arabic web searches, the most searched queries on the Arabic search engine Araby showed that Arab users conform to other internet users in using very short and rather unspecific search queries [Jansen *et al.*, 2000]. Search operators (e.g. Boolean operators) were not used in search queries.

3.6 Processing queries

To collect search results from all tested search engines at almost the same time, the queries were processed on a single day one query at a time on all engines with a minimal time interval. This eliminates the possibility of index changes over the tested search engines while processing the single queries which may give one engine an advantage over the others. Results lists were then saved as HTML pages on a local drive.

3.7 Experimental design

The experimental design in this test is based on the repeated-measures design presented in [Tague-Sutcliffe, 1992] and used in [Griesbaum, 2004].
The jurors had to evaluate the top ten search results (from one to ten) for a single query presented by each web search engine without knowing the source of the results to avoid bias caused by users' preferences of a particular search engine they are familiar with.
The second task was to evaluate the search results descriptions of a single different query on the five tested engines. In this part the sources of the results were known to jurors, as they evaluate the descriptions on the locally saved results pages which are identical to the original results pages delivered by the search engines when queries were executed.

3.8 Data collection

The initial design was to collect data from jurors in a laboratory environment on printed evaluation forms. This design faced however difficulties and was replaced with an online survey design as detailed later in the "Pre-Test" section. Data was collected using an online survey service in digital formats which enable different analyses.

3.9 Data analysis

To obtain a binary relevance judgment, not found documents were added to "not relevant" documents in the results evaluation calculations. Results with no descriptions are also considered "seems not relevant".
Using the collected data, the performance of the five tested search engines was evaluated based on the top ten precision. Macro- and micro-precision for the top ten search results were calculated to evaluate the retrieval performance.
Micro-precision values are also calculated for the top ten results descriptions to analyze the conformity of search results and their descriptions by comparing these values and applying measures presented in [Lewandowski, 2008].

3.10 Presenting results

The test motivation, design and methodology were detailed in the previous sections, the test results are analyzed in a dedicated section and the conclusions section gives a summary of the conducted research and future directions for research based on this work.
The complete work is submitted by the first author as a Master thesis at the University of Hildesheim.

4 Performing the test

4.1 Pre-Test

To examine the initial test design, a pre-test was conducted on 03. April 2009 where six participants executed searches for given queries (one by each juror) with the five tested search engines. The search results pages where recorded and the users judged them based on the results descriptions and subsequently, based on the full result documents. The judgments were given on a printed evaluation form.
This design, however, faced the following main problems:

- It was extremely difficult to plan the test timing to suit all users who do not participate as a part of a university course or a job task.
- The pre-test users found the test tasks complicated, confusing and tiring.
- An extra fatigue effect surfaced as a result of the unreliable internet connection on the test location.

All these problems showed that a laboratory test will not be useful or can not be conducted at all at the time and place initially planned. To avoid the disadvantages of the test location, an alternative solution was to involve Arab jurors geographically distributed over multiple countries by performing the test online as shown in the next section.

4.2 Test

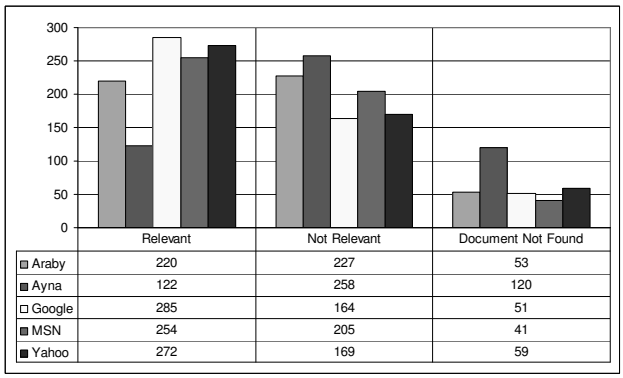
All queries were processed on 11. April 2009 in Germany. Results lists and results were saved locally for documentation and prepared on extra web pages for the evaluation process. Jurors only had to visit given links and evaluate the delivered pages digitally on the provided online form.
The responses collection for online surveys was open in the period from 14. April to 12. May 2009. This long period of time can cause variations in the evaluation process due to the highly dynamic nature of the web; it was, however, needed to allow the large number of jurors to find a suitable time slot in their specific location. This effect can be avoided by obligating the participants to work on a certain date, which rises however the questions about users' motivation.
The collected digital data was relatively easy to analyze and process.

[1] Officially replaced on 03.06.09 with a new search service from Microsoft (www.bing.com)

5 Results

5.1 Number of relevant result documents

The first information that can be obtained about the tested search engines from the evaluation data is the number of relevant result documents; this is displayed in (Figure 1).



	Relevant	Not Relevant	Document Not Found
Araby	220	227	53
Ayna	122	258	120
Google	285	164	51
MSN	254	205	41
Yahoo	272	169	59

Figure 1: Number of relevant result documents

Google retrieved the largest amount of relevant documents followed by Yahoo then MSN. The native Arabic search engine Araby came in the fourth place with a clear distance to Google. A large gap was found between those four search engines and the other Arabic search engine Ayna, which came in the last place.

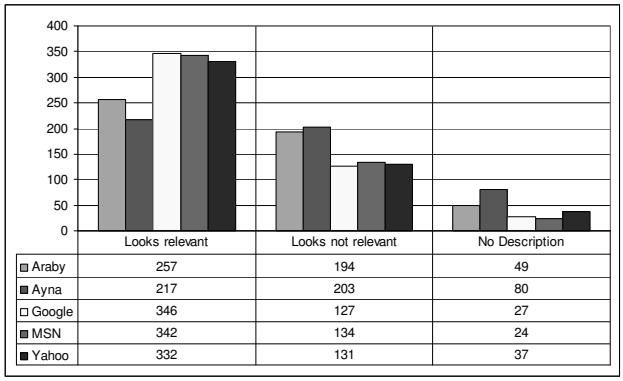
Another finding from the last figure is that both Arabic search engines retrieved a high proportion of absolutely "not relevant" documents (only documents judged as "not relevant" excluding the "not found" documents).

MSN with 8.2% of its results delivered the least search results pointed to lost documents (dead links), followed by Google with 10.2% then Araby which was, with 10.6%, better than Yahoo with 11.8% of dead links in its results lists. Again Ayna came in the last place with more than the double that ratio of all other engines.

These numbers can give an idea on the up to datedness of the search engines indices to a certain extent, but can also be influenced by many factors.

5.2 Number of relevant descriptions

The number of documents among the top ten hits which seemed to be relevant according to their snippet from the five search engines is shown in Figure 2.



	Looks relevant	Looks not relevant	No Description
Araby	257	194	49
Ayna	217	203	80
Google	346	127	27
MSN	342	134	24
Yahoo	332	131	37

Figure 2: Number of relevant results description

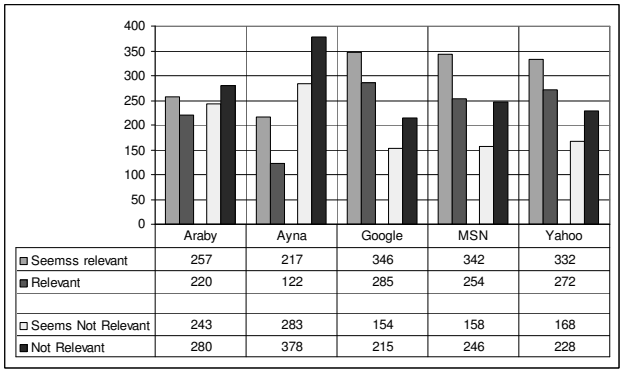
The best judged descriptions were from Google which delivered the highest ratio of relevant descriptions (69.2%). Close after Google was MSN with 68.4% then Yahoo with 66.4% of positively judged descriptions. A clear distance separated these from Araby with 51.4% and Ayna with 43.4%.

The proportion of 40.6% of irrelevant descriptions in the results list of Ayna means that a user may ignore up to this amount of top ten search results because of their descriptions. 38.8% of the descriptions delivered by Araby also gave a negative idea about the results described. With 26.8% for MSN, 26.2% for Yahoo and 25.4% for Google the international search engines gave a lower chance for bypassing results from the first look at their descriptions.

MSN tried to describe the most delivered results out of which 4.8% did not have a description. Google failed similarly to deliver descriptions to 5.4% of presented results and a higher proportion was by Yahoo at 7.4%. Even with 9.8% of results without descriptions, Araby was better than Ayna which delivered 16% of its top ten search results without any description.

5.3 Descriptions-Results conformity

To evaluate how good a search engine can form results descriptions, a comparison between results and results descriptions is needed.



	Araby	Ayna	Google	MSN	Yahoo
Seemss relevant	257	217	346	342	332
Relevant	220	122	285	254	272
Seems Not Relevant	243	283	154	158	168
Not Relevant	280	378	215	246	228

Figure 3: Descriptions-result s conformity

The comparison displayed in (Figure 3) sums not found results to "not relevant" ones and not available descriptions to descriptions "seem not relevant". It shows that all tested search engines presented more relevant descriptions than real relevant results.

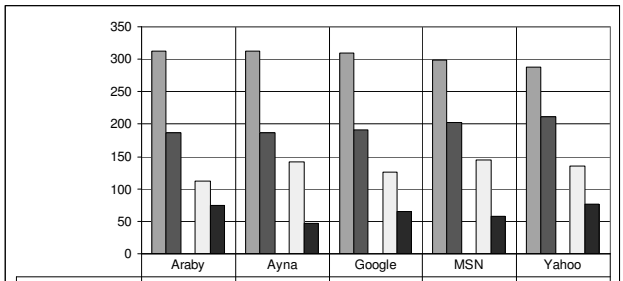
Although the number of relevant descriptions delivered by Araby was lower than the relevant descriptions delivered by the three international search engines, Araby had the lowest difference of 7.8% between the counts of relevant descriptions and relevant results.

For Google, there were 9.7% more relevant snippets than relevant documents, 9.9% for Yahoo and 14.8% for MSN. Ayna exhibited the largest difference of 28% between the numbers of relevant descriptions in comparison with relevant results.

As the relevance judgment of results and results descriptions for each query was done by two different jurors, these results can be influenced as discussed in [Griesbaum, 2004] by formal and contextual variations in the descriptions presentation and by preference factors.

A high number of relevant descriptions does not mean necessarily that they correctly describe the real results and that users could depend on these descriptions to visit relevant results and avoid irrelevant ones. Figure 4 shows the number of documents for which the relevance judgment based on snippet and full document was equal or different. A high judgment consistence of description-result pairs

means a well forming of search results descriptions for both "relevant" and "not relevant" results.



	Araby	Ayna	Google	MSN	Yahoo
Equally judged	313	313	309	298	288
Differently judged	187	187	191	202	212
Description Better	112	141	126	145	136
Result Better	75	46	65	57	76

Figure 4: Description-result pairs judgment comparison

The two native Arabic search engines were equally the best in this regard by presenting the highest ratio of consistent description-result pairs. 62.6% of top ten descriptions delivered by each of Araby and Ayna were identically judged as their respective results. The second conformity level was achieved by Google with 61.8% of presented description-result pairs, followed by MSN with 59.6% and lastly Yahoo with 57.6%.

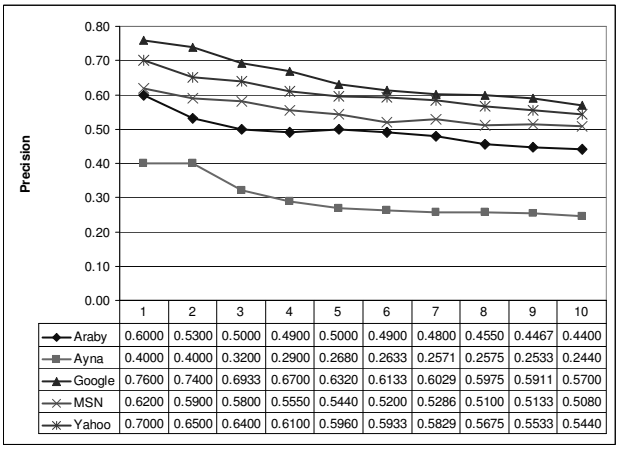
Another output of the last figure is how frequent do the tested search engines tend to present irrelevant results with descriptions that reveal to the user that they can be relevant.

75.4% of Ayna's not matching descriptions gave a better image of the results than they really were, followed by MSN with a close frequency of 71.8% then came Google with 65.9%, closer to Yahoo with 64.2%.

Araby was the search engine that provided the least descriptions which guided the users to results not actually of the same relevance with 59.9% of the total inconsistent descriptions.

5.4 Results mean average precision (micro-precision)

The recall/precision graph plotted in (Figure 5) shows the precision average values for each search engine at the respective rank for the top ten results.



	1	2	3	4	5	6	7	8	9	10
Araby	0.6000	0.5300	0.5000	0.4900	0.5000	0.4900	0.4800	0.4550	0.4467	0.4400
Ayna	0.4000	0.4000	0.3200	0.2900	0.2680	0.2633	0.2571	0.2575	0.2533	0.2440
Google	0.7600	0.7400	0.6933	0.6700	0.6320	0.6133	0.6029	0.5975	0.5911	0.5700
MSN	0.6200	0.5900	0.5800	0.5550	0.5440	0.5200	0.5286	0.5100	0.5133	0.5080
Yahoo	0.7000	0.6500	0.6400	0.6100	0.5960	0.5933	0.5829	0.5675	0.5533	0.5440

Figure 5: Precision graph for the top ten hits

This describes how relevant results were distributed in the top ten results lists.

Google achieved the best values at the top three results ranks, which are usually the most seen by users [Jansen *et al.*, 2000]. 76% of the results delivered by Google at the first ranking place for all queries were judged as relevant, where the average of the sum of relevant results for the top ten results was 57% (rank 10).

The next search engine on the top three ranks was Yahoo with 70% for the first rank and a precision closer to Google at the last rank with 54.4%. Then came MSN which reached 62% at the first rank and 50.8% at the last.

The Arabic search engine Araby reached precision values not far from MSN especially at the first rank with 60% of relevant results at the first place in the results list for the fifty queries. However, the later ranks showed higher differences especially compared to Google. The overall precision at the tenth rank for Araby was at 44%.

Ayna delivered results at the top of results list with lower precision than the results at the last rank of all other engines. The average precision for the first rank results of fifty queries was 40% only. Ayna exhibited a drop in precision after the second position. The precision at the tenth position was merely 24.4%.

The mean average precision values of the top one to ten results for the five tested search engines are shown in (Table 1).

Search Engine	Mean Average Precision
Araby	0.49
Ayna	0.30
Google	0.65
MSN	0.55
Yahoo	0.60

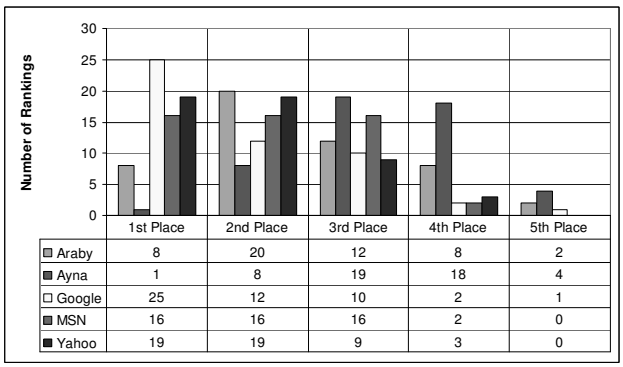
Table 1: Mean average precision

Although Araby performed much better than Ayna, both search engines could not reach an acceptable precision value of 50%, where all other engines stayed above this value even at the last ranking places.

5.5 Answering queries (macro-precision)

To explore which search engine dealt best with every query of the fifty used in the test, the macro-precision is observed. The precision values from all tested search engines for every single query are compared and the engines are ranked accordingly. Search engines with equal precision values for the same query are ranked equally to avoid preferences. The rankings sum comparison should give an overall macro-precision performance view.

These ranking frequencies are displayed in (Figure 6). The numbers show how many times each search engine occupied which ranking place in the comparison.



	1st Place	2nd Place	3rd Place	4th Place	5th Place
Araby	8	20	12	8	2
Ayna	1	8	19	18	4
Google	25	12	10	2	1
MSN	16	16	16	2	0
Yahoo	19	19	9	3	0

Figure 6: Answering queries (macro-precision)

The interpretation of the results reveals the over performance of Google with 25 times ranked at the first place. This means that Google achieved the best top ten precision for 50% of all processed queries. Moreover, it maintained its position in the first two ranks for 74% of these queries and had the worst precision for one single query.
Yahoo was equally ranked at the first and second places for 38% of processed queries at each rank, but was never at the last place. MSN came then with distributed precision performance equally at the first three rankings with 16 times at each rank.
Araby was the first search engine for only 8 times and performed at the second and third levels for 64% of the processed queries. It was the worst engine with two queries. Ayna, on the other hand, was mostly at the third and fourth places and could reach the first place for only one query.

5.6 Number of answered queries

To compare the degree to which each search engine could be helpful for the user, the number of answered queries (queries with at least one retrieved result judged as relevant) is calculated.
(Table 2) shows that Google and Yahoo could answer all processed queries by retrieving at least one relevant result in the top ten results list. The top ten results from MSN for the two queries "Visual illusion" and "Arabic language" included no relevant results.

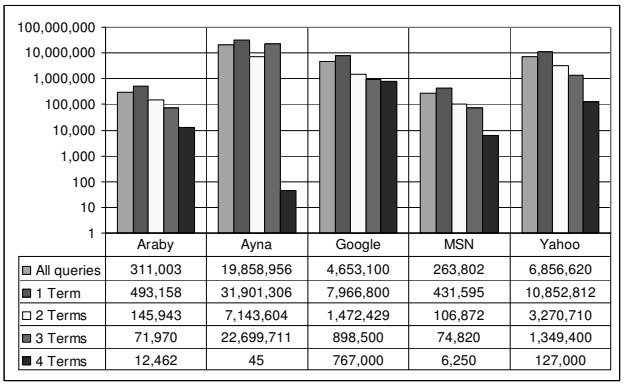
Search Engine	Answered queries	Not Answered queries
Araby	46	4
Ayna	39	11
Google	50	0
MSN	48	2
Yahoo	50	0

Table 2: Number of answered queries

Araby could not answer the four queries "Olympiad", "Obama", "Visual Illusion" and "Sayings". Although these queries can give an impression that the search engine was of no use for the users who entered these unanswered queries (considering that they only see the top ten results), the findings could be influenced by the subjective judgment and their acceptance can be limited. The results of Ayna seemed, however, clearly disappointing as it performed the worst with 22% of not answered queries.

5.7 Number of retrieved documents

Although a detailed estimation of indices sizes for the tested search engines is not within the scope of this work, a general idea about these indices can be obtained from analyzing the amount of retrieved documents reported by the engines when processing the queries. The average counts of results delivered by each search engine for all queries and classified by query terms count are displayed in (Figure 7).
Ayna reported the largest amount of results for each search query even when it performed the worst as seen in all previous evaluation measures. For the search query "Newspapers" for example, Ayna delivered over 33 Millions of results with a top ten precision value of 0 (i.e. the query was not answered). This may question the indexing method and the retrieval algorithm of this search engine, as presenting over millions of irrelevant results can be a sign of an essential index problem or an improper search algorithm. Yahoo and Google delivered a large amount of results in comparison to Araby and MSN.



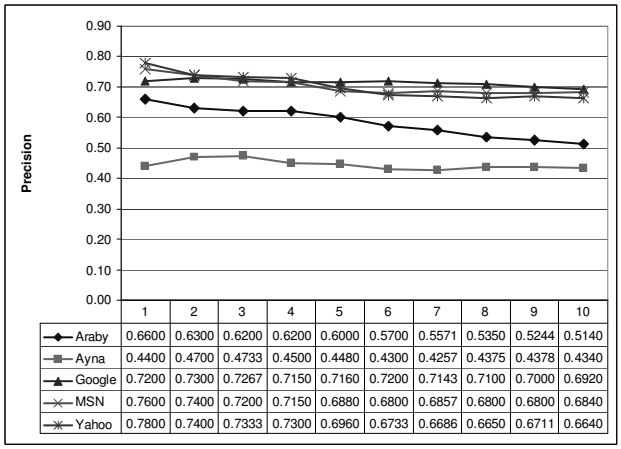
	Araby	Ayna	Google	MSN	Yahoo
All queries	311,003	19,858,956	4,653,100	263,802	6,856,620
1 Term	493,158	31,901,306	7,966,800	431,595	10,852,812
2 Terms	145,943	7,143,604	1,472,429	106,872	3,270,710
3 Terms	71,970	22,699,711	898,500	74,820	1,349,400
4 Terms	12,462	45	767,000	6,250	127,000

Figure 7: Mean number of retrived results

The decrease in average results count for multiple terms queries is clear on all search engines except for Ayna which showed no consistent behavior.

5.8 Descriptions mean average precision

Search results descriptions are evaluated in this work for their importance for users in the decision making to visit a retrieved result. The recall/precision graph of the top ten results descriptions at each result ranking is shown in (Figure 8).



	1	2	3	4	5	6	7	8	9	10
Araby	0.6600	0.6300	0.6200	0.6200	0.6000	0.5700	0.5571	0.5350	0.5244	0.5140
Ayna	0.4400	0.4700	0.4733	0.4500	0.4480	0.4300	0.4257	0.4375	0.4378	0.4340
Google	0.7200	0.7300	0.7267	0.7150	0.7160	0.7200	0.7143	0.7100	0.7000	0.6920
MSN	0.7600	0.7400	0.7200	0.7150	0.6880	0.6800	0.6857	0.6800	0.6800	0.6840
Yahoo	0.7800	0.7400	0.7333	0.7300	0.6960	0.6733	0.6686	0.6650	0.6711	0.6640

Figure 8: Precision graph for the top ten descriptions

All the tested search engines reached better precision values for descriptions than for results themselves (Figure 5) at all cut-off values except for Google at the first two rankings where the average precision for the search results was higher than the average for their descriptions.
To compare the overall precision performance of the results descriptions, the mean average precision for the top one to ten results descriptions from the five search engines is displayed in (Table 3).

Search Engine	Mean Average Precision
Araby	0.49
Ayna	0.30
Google	0.65
MSN	0.55
Yahoo	0.60

Table 3: Mean average precision for the top one to ten results descriptions

5.9 Descriptions-Results comparison

To explore the search engines performance differences in retrieving search results and presenting these results, the evaluations of results and descriptions are compared using measures introduced by Lewandowski [2008].

Mean distance deviance

The distance deviance $DRdist_n$ shows how precision of search results vary from the precision of results description, where n is the number of results or descriptions observed. The following table shows the mean values:

Search engine	$DRdist_{10}$
Araby	0.09
Ayna	0.15
Google	0.08
MSN	0.16
Yahoo	0.10

Table 4: Mean distance deviance of top ten descriptions and results

MSN and Ayna showed the largest difference between descriptions and results precision then Yahoo and Araby. Google had the lowest average precision difference.

We compare the individual description-result pairs on the basis of absolute evaluation values for results and descriptions displayed in (Table 5).

Description	Result	Araby	Ayna	Google	MSN	Yahoo
Relevant	Relevant	146	77	221	198	197
(a)						
Relevant	Not relevant	113	124	127	146	137
(b)						
Not relevant	Relevant	76	123	66	58	77
(c)						
Not relevant	Not relevant	165	158	86	98	89
(d)						
Total number of documents (e)		500	500	500	500	500

Table 5: Individual evaluation counts for description-result pairs

Dividing the pair counts (a, b, c, d) by the total number of documents (e), the precision-result comparison measures can be calculated as shown in (Table 6).

Comparison measure	Araby	Ayna	Google	MSN	Yahoo
Description-result precision (a/e)	0.29	0.15	0.44	0.40	0.39
Description-result conformance (a+d)/e	0.62	0.47	0.61	0.60	0.57
Description fallout (c/e)	0.15	0.25	0.13	0.12	0.15
Description deception (b/e)	0.33	0.32	0.17	0.20	0.18

Table 6: Description-result comparison measures

The best case is when the search engine delivers relevant documents with descriptions that make them appear relevant to the user.

Description-result precision

Google had the highest description-result precision (super precision) followed by MSN then Yahoo. Araby followed with a clear gap, where Ayna was far behind all other engines.

The recall/precision graph for relevant results described with relevant descriptions is plotted in (Figure 9).

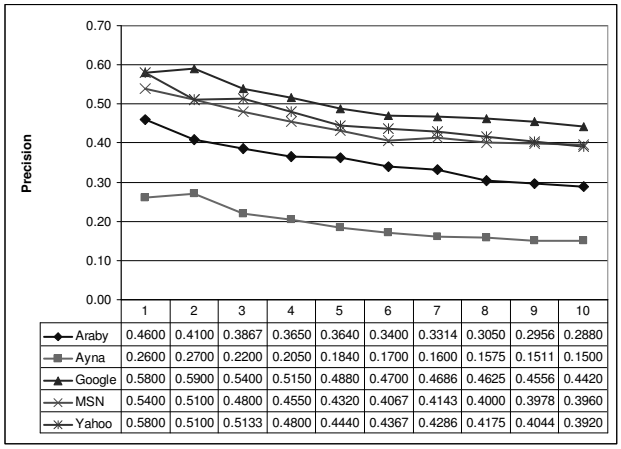


Figure 9: Precision graph for the top ten relevant results and descriptions

Yahoo performed the best for example at the first rank where Google leads with clear difference at the second rank. MSN kept close to the other two engines on all ranks. Araby precision was clearly lower than the three international search engines, and Ayna, which performed at the second rank better than the first rank, was far behind all other engines.

Description-result conformance

From (Table 6) one can see that Araby performed the best by giving the highest amount of "right" described results followed by Google and MSN with small differences, and then came Yahoo followed by Ayna at the last place describing less than a half of delivered results correctly.

Description fallout

Considering that a user may not visit a search result if its description seemed to be irrelevant, the description fallout measures the chance of missing relevant results because of their descriptions.

Most of the tested search engines performed very closely in this regard except for Ayna which described 25% of its relevant search results with seemingly irrelevant descriptions.

Description deception

A high value of description deception can show that the search engine does not provide proper descriptions for irrelevant retrieved results and may cause a frustrating impression to the user who feels misled by the search engine.

Google performed the best in this regard by providing the least amount (17%) of irrelevant results associated with descriptions that let them look relevant. The next search engines were Yahoo then MSN with close values (18% and 20% respectively). Clearly, both Arabic search engines performed worse with 32% of results with misleading descriptions for Ayna and 33% for Araby.

6 Conclusions

As overall result, it can be concluded that Google was the best search engine in all measures except for the number of not found documents, number of results with no descriptions, descriptions fallout and the conformance of results and descriptions.

MSN and Yahoo exchanged the second and third ranking places with regards to most evaluation measures except for the number answered queries where they performed equally and for the number of not found document and the conformance of results and descriptions where Yahoo fell back to the fourth rank.

Moreover, MSN performed best in terms of not found documents count, count of results with no descriptions and description fallout.

Although Araby was mostly on the penultimate place, it showed now significant precision difference to MSN, and delivered the best conformance of results and their descriptions, performed better than Yahoo in terms of not found documents count and was equal to it in description fallout. The only last place given to Araby was in description deception.

The underperformance of Ayna was a remarkable trouble sign. The search engine with the large promotion campaign seemed to suffer from very serious problems in both its indexing and searching algorithms and it obviously would need substantial improvement.

This test found that there is mostly no significant reason to prefer Google to Yahoo in terms of search performance in Arabic language. One should however keep in mind that Yahoo does not offer an Arabic interface (by the time of the test) which can affect its acceptance. Arab users may also still consider MSN as a potential alternative search engine especially when interested in particular performance aspects.

The more important finding of the test is that both tested native Arabic search engines could not proof their ability to compete as a local alternative to international search services. Even when Araby had some good results, a wide space for improvement still exists.

The results of this state of the art work can be considered for further evaluations and research of Arabic search engines, particularly with the absence of similar published studies for this language.

References

[Abdelali *et al.*, 2004] Ahmed Abdelali, Jim Cowie, and Hamdy S. Soliman. Arabic Information Retrieval Perspectives. In: *Proceedings of JEP-TALN 2004 Arabic Language Processing*, Pages 19-22. April. 2004.

[Gordon and Pathak, 1999] Michael Gordon and Praveen Pathak. Finding Information on the World Wide Web: the Retrieval Effectiveness of Search Engines. *Information Processing & Management*, 35:141-180.

[Griesbaum 2004] Joachim Griesbaum. Evaluation of three German search engines: Altavista.de, Google.de and Lycos.de. In: *Information Research*, 9(4). <http://informationr.net/ir/9-4/paper189.html> (Accessed 10.08.09)

[Hammo 2009] Bassam H. Hammo: Towards enhancing retrieval effectiveness of search engines for diacritisized Arabic documents. In: *Information Retrieval*, 12(3):300-323, June 2009, Springer, Netherlands.

[Hawking *et al.*, 2001] David Hawking, Nick Craswell, Peter Bailey and Kathleen Griffiths. Measuring Search Engine Quality. In: *Information Retrieval*, 4(1):33-95 Springer, Netherlands.

[Jansen *et al.*, 2000] Bernard J. Jansen, Amanda Spink and Tefko Saracevic. Real life, real users, and real needs: a study and analysis of user queries on the Web. In: *Information Processing & Management*, 36(2):207-227.

[Lazarinis, 2007] Fotis Lazarinis. Web retrieval systems and the Greek language: do they have an understanding? In: *Journal of Information Science*, 33(5):622-636, Sage Publications Inc.

[Lazarinis, 2009] Fotis Lazarinis, Jesus Vilares, John Tait and Efthimis Efthimiadis. Current research issues and trends in non-English Web searching. In *Information Retrieval*, 12(3):230-250, June. 2009.

[Lewandowski, 2008] Dirk Lewandowski. The Retrieval Effectiveness of Web Search Engines: Considering Results Descriptions. In: *Journal of Documentation*, 64(6):915-937.

[Lewandowski and Höchstötter, 2008] Dirk Lewandowski, and Nadine Höchstötter. Web Searching: A Quality Measurement Perspective. In: A. Spink and M. Zimmer (Editors.): *Web Searching: Multidisciplinary Perspectives*. Pages: 309-340, Springer, Berlin.

[Mandl, 2008] Thomas Mandl. Recent Developments in the Evaluation of Information Retrieval Systems: Moving Toward Diversity and Practical Applications. In *Informatica - An International Journal of Computing and Informatics*, 32:27-38.

[Moukdad, 2002] Haidar Moukdad: Language-based retrieval of Web documents: An analysis of the Arabic-recognition capabilities of two major search engines. In: *Proceedings of the 65th Annual Meeting of the American Society for Information Science and Technology*, 18-21. November. 2002, Philadelphia, PA. Medford: Information Today, Poster p. 551.

[Moukdad, 2004] Haidar Moukdad. Lost in Cyberspace: How do search engines handle Arabic queries? In: Access to Information: Technologies, Skills, and Socio-Political Context. Proceedings of the 32nd Annual Conference of the Canadian Association for Information Science, Winnipeg, 3-5. June. 2004.

[Moukdad and Large, 2001] Haidar Moukdad, and Andrew Large. Information retrieval from full-text Arabic databases: Can search engines designed for English do the job? Libri 51(2):63-74.

[Schamber, 1994] Linda Schamber. Relevance and information behavior. *Annual Review of Information Science and Technology*. 29:3-48.

[Tague-Sutclife, 1992] Jean Tague-Sutclife. The Pragamatics of Information Retrieval Experimentation, Revisited. In: *Information Processing & Management*, 28(4): 467-490, Elsevier.

[Womser-Hacker, 1989] Christa Womser-Hacker. Der PADOK Retrievaltest. Zur Methode und Verwendung statistischer Verfahren bei der Bewertung von Information-Retrieval-Systemen. Hildesheim, Georg Olms.

Ein Polyrepräsentatives Anfrageverfahren für das Multimedia Retrieval

David Zellhöfer und Ingo Schmitt
Brandenburgische Technische Universität Cottbus
Lehrstuhl für Datenbank- und Informationssysteme
david.zellhoefer@tu-cottbus.de

Abstract

Multimediale Dokumente sind durch eine Vielzahl an Aspekten gekennzeichnet, die während einer Suche relevant werden können. Diese Aspekte werden selten ganzheitlich durch Retrieval-Modelle betrachtet, weshalb in dieser Arbeit ein Anfragemodell vorgeschlagen wird, welches das polyrepräsentative Prinzip umsetzt. Konzeptionell gesehen basiert dieses Anfragemodell auf Ergebnissen der Quantenlogik.

Das Modell bietet dabei Werkzeuge an, um die Relevanz eines Dokuments gegenüber einer Cognitive Overlap aller Dokumentenrepräsentation zu berechnen. Durch den Einsatz des vorgestellten Verfahrens wird es erstmals möglich, die Cognitive Overlap auf Grundlage der Gesetze der Booleschen Algebra zu modellieren. Zusätzlich wird ein Relevance-Feedback-Verfahren zur weiteren Personalisierung der Ergebnisdokumente präsentiert. Dieses Verfahren nutzt als Eingabe qualitative Präferenzbewertungen auf den Ergebnisdokumenten, welche den Nutzer davon befreien, sich mit dem zugrundeliegenden Anfragemodell oder der Gewichtung der einzelnen Repräsentationen auseinanderzusetzen.

Abschließend zeigt ein Experiment den Nutzen des vorgestellten Anfragemodells.

1 Einleitung

Mit der steigenden Verfügbarkeit multimedialer Inhalte wie Videos, Bildern, Musik oder Rich-Media-Applikationen wird es notwendig, geeignete Suchmechanismen anzubieten. Multimedia-Dokumente sind durch verschiedene Aspekte charakterisiert, die während eines bestimmten Suchbedürfnisses oder -kontexts relevant werden können. Die Suche wird außerdem dadurch erschwert, dass die Vorlieben für einzelne Aspekte seitens der Nutzer je nach Szenario teils stark variieren. Während zu einem Zeitpunkt die Farbstimmung eines Bildes besonders wichtig erscheint, kann dies zum nächsten Zeitpunkt ausschließlich der Hersteller des Dokuments sein. Aus diesem Beispiel wird klar, dass die Relevanzbewertung eines Dokuments selten statisch ist.

Die bereits erwähnte Vielzahl an Aspekten von Multimedia-Dokumenten wird auch in der internen Repräsentation innerhalb des Retrieval-Systems widergespiegelt. Diese werden in der Regel in atomare Medientypen wie Töne oder Bilder aufgespalten. Die dabei entstehenden Fragmente können dann mittels Low-Level-Features (LLF) wie Farbe oder Textur, High-Level-Features (HLF) wie Annotationen oder aber mittels Metadaten (MD) wie Hersteller oder Produktionsdatum beschrieben werden.

Aktuelle Verfahren des Multimedia Retrievals basieren meist auf Low-Level-Features, da diese im Gegensatz zu den semantisch aussagekräftigeren High-Level-Features maschinell extrahierbar sind. Momentan erreichen diese Ansätze jedoch eine „glass ceiling“, die nicht durch die Hinzunahme weiterer LLF überwunden werden kann [Aucouturier and Pachet, 2004]. Zur Überwindung dieser Barriere werden in der Regel Tags oder Annotationen zusätzlich durch das Retrieval-System ausgewertet. Weitere verfügbare Metadaten wie z.B. Exif[1] werden oft vernachlässigt, da sie sich vor allem gut in relationalen Datenbanken (DB) ablegen lassen und dort strukturiert ausgewertet werden können.

Diese Arbeit widmet sich dem diskutierten Problem vom kognitiven Standpunkt aus. Relevanzbewertungen bezüglich eines Suchbedürfnisses werden nicht durch eine beliebige Kombination aus LLF oder HLF getätigt. Vielmehr wird Relevanz als eine *cognitive overlap* [Larsen *et al.*, 2006] aller vorliegender Repräsentationen eines Dokuments modelliert. Hierbei wird ignoriert, ob diese Repräsentationen einen Ursprung im Information Retrieval (IR) oder im DB-Bereich haben. Um diesen Ansatz zu motivieren, wird das polyrepräsentative Prinzip umrissen. Das darauf folgende Kapitel diskutiert ein polyrepräsentatives Anfragemodell, welches auf Erkenntnissen der Quantenlogik basiert. Dieses Modell wird dazu verwendet, Multimedia-Dokumente und die Suche nach ihnen holistisch abzubilden. Des weiteren wird ein Verfahren zur Personalisierung der Ergebnisse vorgeschlagen. Das präsentierte Modell kann dabei vor allem als Antwort auf die offene Frage nach einem Mechanismus zur Bildung einer Cognitive Overlap mittels strukturierter Anfragen [Larsen *et al.*, 2006] gesehen werden. Kapitel 4 zeigt erste Experimente auf Grundlage des theoretischen Modells. Den Abschluss der Arbeit bildet eine Zusammenfassung der Ergebnisse und der Ausblick auf offene Forschungsfragen.

2 Das Polyrepräsentive Prinzip im Information Retrieval

Das Prinzip der Polyrepräsentation wurde als kognitives Modell im IR vorgeschlagen [Ingwersen and Järvelin, 2005]. Es basiert auf der Annahme, dass verschiedene Repräsentationen eines Dokuments dazu ausgenutzt werden können, eine *cognitive overlap* [Larsen *et al.*, 2006] zu bilden. Diese Überlappung kann durch eine scharfe Schnittmenge zwischen funktional und kognitiv unterschiedlichen Repräsentationen eines Dokuments dargestellt werden, wie

[1] Exchangeable Image File Format; http://www.exif.org

Abbildung 1 zeigt. Dabei versteht man unter *funktional* unterschiedlichen Repräsentationen Titel, Abstracts, Volltexte etc., die von einem Akteur erstellt wurden. Dem gegenüber stehen *kognitiv* unterschiedliche Repräsentationen, die sich nach dem aktuellen Suchbedürfnis, persönlichen Vorlieben oder Interpretationen durch Dritte, wie indizierende Bibliothekare richten[2]. Basierend auf dieser Cognitive Overlap kann die *probability of relevance* (POR) eines Dokuments bezüglich einer Anfrage – oder allgemeiner gesprochen: eines Informationsbedürfnisses – berechnet werden.

Aufgrund der Kombination verschiedener funktionaler und kognitiver Repräsentationen wird dabei angenommen, dass die intrinsische Unsicherheit von Relevanzbewertungen im IR verringert werden kann, was letztendlich zu einer Erhöhung der Retrievalqualität führt [Ingwersen and Järvelin, 2005]. Diese Hypothese wird auch durch weitere Arbeiten unterstützt [Skov *et al.*, 2004].

Zwei wesentliche Erkenntnisse der letztgenannten Arbeit betreffen die Strukturiertheit der verwendeten Anfrage und die Menge der verwendeten Repräsentation, welche die Cognitive Overlap bilden. Hierbei kann festgehalten werden, dass eine große Anzahl an kognitiv und funktional unterschiedlichen Repräsentationen die Präzision des IR-Systems erhöht. Allerdings muss angemerkt werden, dass der Grad an Strukturiertheit, welcher in der Anfrage vorliegt, ebenfalls einen wesentlichen Einfluss auf die Präzision des Systems hat, da sie letztendlich den Aufbau der Cognitive Overlap bestimmt. Anfragen können dabei von *hochstrukturiert*, d.h. sie greifen auf Boolesche Junktoren zurück, bis *unstrukturiert*, wie z.B. Bag-of-Word-Anfragen, reichen. Laut [Skov *et al.*, 2004] muss hochstrukturierten Anfragen der Vorzug gegeben werden, da sie niedriger strukturierten Anfragen im Rahmen der Retrievalqualität überlegen sind. Diese Erkenntnis spiegelt sich auch in den Ergebnissen von [Hull, 1997; Turtle and Croft, 1991] wieder. Zusammenfassend kann gesagt werden, dass eine hohe Anzahl an Repräsentationen eines Dokuments in Kombination mit einer hochstrukturierten Anfrage die Retrievalqualität deutlich erhöht.

Obwohl die bisherigen experimentellen Ergebnisse des polyrepräsentativen Prinzips vielversprechende Resultate aufweisen, wurde bis dato kein Framework vorgeschlagen, welches es ermöglicht, Cognitive Overlaps hochstrukturiert zu modellieren. Im nächsten Abschnitt wird deshalb ein solches Framework diskutiert, welches außerdem die Personalisierung der Überlappung ermöglicht.

3 Ein Polyrepräsentatives Anfragemodell

Multimedia-Dokumente sind intrinsisch polyrepräsentativ aufgrund der Vielzahl an die Relevanz beeinflussenden Aspekten aus denen sie bestehen und mit denen sie in einem Retrieval-System dargestellt werden; zum einen funktional gesehen (LLF, HLF und MD) als auch im kognitiven Bereich, wie z.B. unter Annahme eines bestimmten Suchbedürfnisses mitsamt der kontextabhängigen Präferenzen zwischen den einzelnen Aspekten. Allerdings bleibt die Frage offen, wie alle existierenden Repräsentationen kombiniert werden müssen um eine Cognitive Overlap zu modellieren, d.h., wie eine Anfrage formuliert werden kann, die das subjektive Verständnis von Ähnlichkeit eines Nutzers bezüglich eines Suchziels wiedergibt.

Vorliegende Arbeit aus dem Bereich IR bestimmen diese Anfragen empirisch für eine gegebene Suchdomäne ohne sich jedoch allgemeinen Suchaufgaben zu widmen. Larsen et al. [Larsen *et al.*, 2006] unterstreichen deshalb den Bedarf an weitergehender Forschung zur Bildung von Cognitive Overlaps „depending on domains, media, genre, and presentation styles“. Nichtsdestotrotz gehen sie nicht auf den Einfluss der subjektiven Relevanzwahrnehmung einzelner Nutzer ein. Letztendlich führt diese Subjektivität zu einer unterschiedlichen Ausprägung der Cognitive Overlap obwohl ihre Zusammensetzung ähnlich bleibt. Dies schlägt sich in unterschiedlich starken Einflüssen einzelner Repräsentationen je nach Nutzer oder Anwendungskontext nieder (siehe Abschnitt 3.2).

Ein weiterer Kritikpunkt an [Larsen *et al.*, 2006] ist die Annahme von Booleschen Junktoren zur Verknüpfung der Repräsentationen, was unter Umständen zu einer leeren Ergebnismenge führen kann, wenn kein Dokument die Bedingungen, um in die Cognitive Overlap zu gelangen, erfüllt. Dieses Problem existiert in unserem Ansatz nicht, da eine Ähnlichkeit zur Cognitive Overlap berechnet wird. Das heißt, wenn einzelne Bedingungen nicht durch ein Dokument erfüllt werden, so wird dieses „abgestraft“. Folglich ergibt sich eine weiche Grenzen, weshalb von einem *Penetrable Cognitive Overlap* (PCO) gesprochen werden kann (siehe Abbildung 1).

Frommholz und van Rijsbergen [Frommholz and van Rijsbergen, 2009] erkennen ebenfalls das Problem eines fehlenden Frameworks, welches das polyrepräsentative Prinzip unterstützt. Ihre Arbeit basiert, wie die hier vorgestellte, auf Resultaten der Quantenmechanik und -logik [van Rijsbergen, 2004], diskutiert allerdings nicht die Verwendung von strukturierten Anfragen, um die Cognitive Overlap zu modellieren.

Um strukturierte Cognitive Overlaps zu konstruieren, schlagen wir deshalb das folgende Anfragemodell vor. Das Modell besteht aus einer Anfragesprache, die im kommenden Abschnitt kurz vorgestellt wird, und einem adaptiven Relevance-Feedback-Prozess (RF) [Rocchio, 1971] für die Nutzerinteraktion und Personalisierung (siehe Abschnitt 3.2).

3.1 Verwendung der *Commuting Quantum Query Language* im Modell

Das vorgestellte Verfahren basiert auf der *commuting quantum query language* (CQQL) [Schmitt, 2008], welche sich auf die Quantenlogik [Birkhoff and Neumann, 1936] stützt. CQQL bietet als Anfragesprache die Integration von ähnlichkeitsbasierten IR-Prädikaten und relationalen DB-Prädikaten. Folglich kann sie deshalb sämtliche Zugriffsparadigmen verwenden, die bei der Verarbeitung und Durchsuchung von Multimedia-Dokumenten existieren. Zusätzlich gehorcht CQQL den Gesetzen der Booleschen Algebra und ist deshalb einer hochstrukturierte Anfragesprache, die alle in [Skov *et al.*, 2004; Hull, 1997; Turtle and Croft, 1991] beschriebenen Vorteile in sich vereint. Sie unterstützt damit die Anforderung, dass „structured Boolean-like query configurations will best support polyrepresentation in IR“ [Larsen *et al.*, 2006]. Hierbei muss angemerkt werden, dass die Nutzung einer Booleschen Algebra nicht mit den Nachteilen des Booleschen Retrieval-Modells, wie ungeordneten Ergebnislisten oder dem Verzicht auf Termgewichtung, gleichzusetzen ist. CQQL nutzt Ähnlichkeits- bzw. Wahrscheinlichkeitswerte, um die POR eines Dokuments zu bestimmen. Beispiele für den Einsatz von CQQL zur Konstruktion von Cognitive Overlaps folgen weiter unten.

Um die theoretischen Grundzüge zu umreissen, werden nachfolgend die konzeptionellen Bezüge zwischen Quantenlogik und -mechanik sowie dem polyrepräsentativen

[2] Dabei muss angemerkt werden, dass bereits die Kombination aus einem kontrollierten Vokabular, verschiedenen Stemming-Verfahren o.ä. polyrepräsentativ ist.

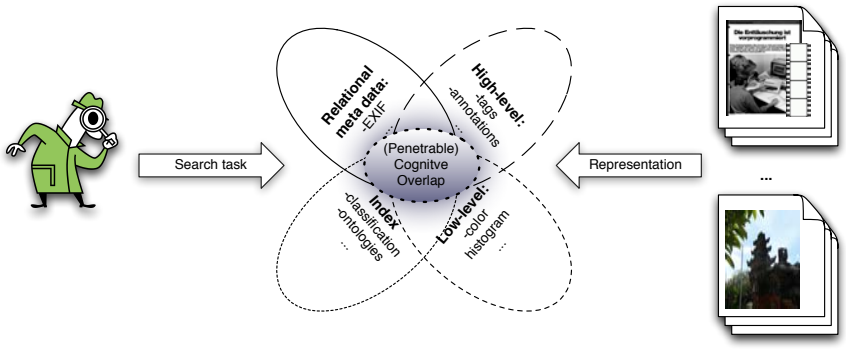


Abbildung 1: Venn-Diagramm verschiedener Dokumentrepräsentationen, die eine Penetrable Cognitive Overlap bilden.

Prinzip skizziert. In der Quantenmechanik kann jeder Zustand eines mikroskopischen Objekts durch einen normalisierten *Zustandsvektor* $|\varphi\rangle$[3] beschrieben werden. Die Zustände werden in einem Hilbertraum $\mathbf{H}$ beschrieben – vereinfacht gesagt: einem reellwertigen Vektorraum mit einem Skalarprodukt. Im vorgestellten Ansatz interpretieren wir einen Zustandsvektor als eine mögliche Repräsentation eines Dokuments, z.B. ein konkretes LLF.

Um den Zustand eines Systems zu messen, nutzen wir *Projektoren*. Ein Projektor $p = \sum_i |i\rangle\langle i|$ ist ein symmetrischer und idempotenter linearer Operator, der über einer Menge von orthonormalen Vektoren $|i\rangle$ definiert wird. Die Multiplikation eines Projektors mit einem Zustandsvektor entspricht der Projektion des Vektors in den entsprechenden Vektorunterraum. In anderen Worten wird die Wahrscheinlichkeit berechnet mit welcher ein Zustandsvektor für einen Aspekt der Cognitive Overlap relevant ist. Die Ähnlichkeit des Projektors p und eines Zustandsvektors $|\varphi\rangle$ ist dabei wie folgt definiert:

$$\langle\varphi|p|\varphi\rangle = \langle\varphi|(\sum_i |i\rangle\langle i|)|\varphi\rangle = \sum_i \langle\varphi|i\rangle\langle i|\varphi\rangle$$

Interpretiert man diese Quantenmessung nun als Wahrscheinlichkeitswert, dann ist dieser Wert gleich der quadrierten Länge des Zustandsvektors nach der Projektion in den Vektorunterraum, welcher durch die Vektoren $|i\rangle$ aufgespannt wird. Außerdem entspricht der Wert aufgrund der Normalisierung dem quadrierten Kosinus des minimalen Winkels zwischen $|\varphi\rangle$ und dem durch p repräsentierten Vektorunterraum. Sollen mehrere Quantensysteme gemessen werden, d.h. mehrere Repräsentation zum Einsatz kommen, wird das Tensorprodukt zu deren Verknüpfung verwendet. Das weitere Vorgehen gleicht dem bereits geschilderten. Für Details, insbesondere zur Verknüpfung einzelner Quantensysteme durch das Tensorprodukt und die daraus folgenden theoretischen Eigenschaften sowie Beweise wird aus Platzgründen auf [Schmitt, 2008] verwiesen. Tabelle 1 fasst die bisherigen Erkenntnisse zusammen.
Aufgrund des Fokus' dieser Arbeit wird für notwendige Normalisierungsschritte und -algorithmen, die vor der Auswertung einer CQQL-Anfrage nötig sind, auf [Schmitt, 2008] verwiesen. Stattdessen wird direkt die resultierende algebraische Auswertung einer CQQL-Anfrage vorgestellt, da sie später für die Messung der POR eines Dokuments notwendig ist.

[3]In Anlehnung an die übliche Schreibweise innerhalb der Quantenmechanik greifen wir hier auf die Dirac-Notation für Vektoren zurück. Bei dieser wird ein Spaltenvektor als $|x\rangle$ angegeben während $\langle y|$ einen Zeilenvektor bezeichnet. $\langle x|y\rangle$ drückt folglich das Skalarprodukt beider Vektoren aus.

Tabelle 1: Gegenüberstellung der Konzepte zwischen CQQL und dem Polyrepräsentativen Prinzip

CQQL	**Polyrepräsentation**
Zustandsvektor	Repräsentation
Projektor	Cognitive Overlap
Quantenmessung	Probability of Relevance bzgl. d. Cognitive Overlap

Es sei $f_\varphi(d)$ die Evaluierung eines Dokuments d bezüglich eines PCO φ und $\varphi_1 \wedge \varphi_2, \varphi_1 \vee \varphi_2$ sowie $\neg\varphi$ mittels logischen Junktoren verbundene Repräsentation (*Prädikate*), welche die PCO modellieren. Ist φ ein atomares Prädikat, so kann $f_\varphi(d)$ direkt in einen numerischen Wert im Intervall $[0;1]$ ausgewertet werden, welcher die POR von d bezüglich der PCO darstellt. Ansonsten werden die CQQL-Prädikaten rekursiv entsprechend der folgenden Formeln ausgewertet:

$f_{\varphi_1\wedge\varphi_2}(d) = f_{\varphi_1}(d) * f_{\varphi_2}(d)$
$f_{\varphi_1\vee\varphi_2}(d) = f_{\varphi_1}(d) + f_{\varphi_2}(d) - f_{\varphi_1}(d) * f_{\varphi_2}(d)$
wenn φ_1 und φ_2 nicht exklusiv sind
$f_{\varphi_1\vee\varphi_2}(d) = f_{\varphi_1}(d) + f_{\varphi_2}(d)$
wenn φ_1 und φ_2 exklusiv sind
$f_{\neg\varphi}(d) = 1 - f_\varphi(d)$.

Hierbei heißt eine Disjunktion *exklusiv*, wenn sie folgende Form hat: $(\varphi_i \wedge \ldots) \vee (\neg\varphi_i \wedge \ldots)$ für ein φ_i. Hier wird deutlich, dass jede Evaluierung mittels einfacher arithmetischer Operationen durchgeführt wird, welche den Gesetzen der Booleschen Algebra gehorchen. Aufgrund der disjunktiven Normalform, die vor der Auswertung erreicht wird [Schmitt, 2008], kann jede Evaluierung durch eine Summe von Produkten atomarer Prädikate auf einzelnen Repräsentationen ausgedrückt werden. Folglich wird die PCO beliebiger Repräsentationen in einer hochstrukturierten Art modelliert und anschließend mittels einfacher arithmetischer Ausdrücke ausgewertet. Das Ergebnis dieser Auswertung ist die POR eines Dokuments im Intervall $[0;1]$.

Aus der Gegenüberstellung der Evaluierungsregeln von CQQL mit Kolmogorovs Axiomen wird die Beziehung zwischen Quantenmessungen und der Wahrscheinlichkeitstheorie deutlich:

$P(X \cap Y) = P(X) * P(Y)$,
$P(X \cup Y) = P(X) + P(Y)$
(für sich gegenseitig ausschließende Ereignisse), und
$P(X \cup Y) = P(X) + P(Y) - P(X \cap Y)$

Hierbei ist P die Wahrscheinlichkeit eines Ereignisses X

oder Y. Die Negation ist analog definiert.

Eine weitere wesentliche Charakteristik von CQQL ist die Einbettung von Gewichten in die Sprache, was die Personalisierung der PCO erleichtert (siehe Abschnitt 3.2). Die Gewichte θ_i dienen in CQQL dazu, unterschiedliche Wichtigkeiten einzelner Prädikate zu bestimmen. Besonders dabei ist, dass diese Gewichte nicht die Gesetze der Booleschen Algebra verletzen, wie das z.B. bei [Fagin and Wimmers, 2000] der Fall ist, die Gewichtungen außerhalb ihrer Anfragesprache ermöglichen.

Gewichte in einer gewichteten CQQL-Anfrage q_θ sind direkt mit Junktoren verknüpft, wie das folgende Beispiel zeigt: $q_\theta = (\varphi_1 \wedge_{\theta 1,\theta 2} \varphi_2) \wedge_{\theta 3,\theta 4} \varphi_3$. Zur Auswertung werden die Gewichte durch Konstanten wie folgt ersetzt:

$$\varphi_1 \wedge_{\theta_1,\theta_2} \varphi_2 \rightsquigarrow (\varphi_1 \vee \neg\theta_1) \wedge (\varphi_2 \vee \neg\theta_2)$$
$$\varphi_1 \vee_{\theta_1,\theta_2} \varphi_2 \rightsquigarrow (\varphi_1 \wedge \theta_1) \vee (\varphi_2 \wedge \theta_2)$$

Ungewichtete Anfragen werden durch das Setzen aller Gewichte auf 1 ermöglicht. Folgende Beispielanfrage soll den Nutzen von Gewichten verdeutlichen.

> q_θ: Finde alle Dokumente, die einem gegebenen Bild Img ähneln und im Jahr 2009 erstellt wurden.

$$q_\theta := (year = 2009) \wedge_{\theta 1,\theta 2} (edges \approx Img.edges \wedge_{\theta 2,\theta 3} colorLayout \approx Img.colorLayout \wedge_{\theta 4,\theta 5} (blue \approx Img.blue \vee_{\theta 6,\theta 7} orange \approx Img.orange))$$

In diesem Beispiel wird das Jahr als Boolesches Prädikat, z.B. aus relationalen Metadaten ausgelesen, angenommen, während die anderen Bedingungen Ähnlichkeitsmaße auf Grundlage von LLF darstellen. Die Gewichte θ_i dienen dabei zur Steuerung des Einflusses der einzelnen Bedingungen auf die Evaluierung von q_θ.

3.2 Nutzerinteraktion zur Personalisierung der Penetrable Cognitive Overlap

Ein Problem, welches häufig bei der Modellierung von Cognitive Overlaps vernachlässigt wird, ist die Subjektivität von Nutzerpräferenzen zwischen den verschiedenen Repräsentationen. Während ein Nutzer sehr empfindlich auf Texturen, die in einem Bild existieren, reagiert, ist es denkbar, dass ein anderer besonderen Wert auf eine bestimmte Farbtönung und vorhandene Annotationen legt. Obwohl durchaus für beide Suchbedürfnisse die PCO durch die gleiche strukturierte Anfrage modelliert werden kann, ist es offensichtlich, dass die Nutzer nicht über die Wichtigkeit der einzelnen Repräsentationen, welche die PCO ausmachen, übereinstimmen werden. Betrachten wir deshalb ein Beispiel:

> Eine multimediale Dokumentensammlung eines Museums soll nach Dokumenten der Renaissance durchsucht werden. Die Sammlung besteht aus digitalisierten Gemälden, Skulpturen, Photographien und Büchern. Der jeweiligen Medientyp eines Dokuments ist neben Attributen wie Hersteller, Herstellungsjahr, historischer Ort etc. in einer relationalen DB abgelegt. Desweiteren stehen LLF und HLF in einem Retrieval-System zur Verfügung.

In diesem Szenario wird der PCO durch verschiedene funktional und kognitiv unterschiedliche Repräsentationen eines Dokuments, beispielsweise das Jahrhundert der Herstellung, den Medientyp oder vom Medientyp abhängige LLF sowie Annotationen, strukturiert formuliert. Eine entsprechende CQQL-Anfrage kann wie folgt aussehen:

$$q := \neg(mediaType = \text{"photography"}) \wedge (century \approx 15) \wedge ((mediaType = \text{"painting"} \rightarrow dominantColor \approx \text{RGB(...)} \wedge ...) \vee (mediaType = \text{"sculpture"} \rightarrow ...) \vee (mediaType = \text{"book"} \rightarrow ...")) \wedge (subject \approx \text{"madonna"} \vee subject \approx \text{"pope"} \vee ...) \wedge ...$$ [4]

Aufgrund der Strukturiertheit von CQQL wird es möglich Implikationen ($\rightarrow$) anzugeben, um beispielsweise spezielle Mengen von Prädikaten anzugeben, die im Falle eines bestimmten Medientyps ausgewertet werden. Diese Möglichkeit widmet sich direkt dem „multi-domain problem“ [Larsen *et al.*, 2006] und ermöglicht es, verschiedene Medientypen innerhalb einer logischen Anfrage zu integrieren. Im gegebenen Beispiel dient beispielsweise das LLF „Dominant Color“ als funktionale Repräsentationen eines Bildes, während unterschiedliche Repräsentationen für andere Medientypen Einsatz finden. Photographien werden von vornherein ausgeschlossen, da sie zur Renaissance nicht existierten.

Obwohl die allgemeine Konstruktion einer Cognitive Overlap für das Renaissance-Suchszenario leicht verständlich ist, kann davon ausgegangen werden, dass Nutzer unterschiedliche, subjektive Präferenzen zwischen den funktionalen und kognitiven Repräsentationen haben. So wird ein Nutzer den Schwerpunkt auf das Jahrhundert legen, während ein anderer besonderes Augenmerk auf bestimmte Farbtöne[5] oder dargestellte Themen legt. Um die PCO entsprechend zu personalisieren bietet sich die Gewichtung der einzelnen Repräsentationen an.

Gewichtungen haben seit den frühen Extended-Boolean-Retrieval-Modellen eine Tradition im IR. Ihr positiver Effekt auf die Nutzerzufriedenheit und die Personalisierung gilt dabei als gesichert [Hull, 1997; Salton *et al.*, 1983; Lee *et al.*, 1993]. Nichtsdestotrotz müssen diese Gewichte auf einer empirischen Basis bestimmt oder explizit durch den Nutzer gesetzt werden. Insbesondere das Letztgenannte stellt eine komplizierte Aufgabe dar, die auch Kenntnisse des zugrundeliegenden Retrieval-Modells voraussetzt.

Als Beispiel für die Integration von gewichteten Repräsentationen in die Cognitive Overlap können [Skov *et al.*, 2004] genannt werden. In ihrem Ansatz werden Gewichte während der Aggregation der verschiedenen Ähnlichkeitswerte verwendet. Konzeptionell ähnelt dieser Ansatz Fagin und Wimmers Vorgehen [Fagin and Wimmers, 2000], die ebenfalls Anfragelogik und Gewichtung trennen. Im Gegensatz dazu integriert CQQL die Gewichtung in die Logik, was die Kompatibilität mit der Booleschen Algebra garantiert und verhindert, dass sich die Semantik der Anfrage durch die Gewichtung verändert. Übertragen auf das letzte Beispiel führt das zu folgender Modifikation der Anfrage:

$$q_\theta := \neg(mediaType = \text{"photography"}) \wedge_{\theta_1,\theta_2} (century \approx 15) \wedge_{\theta_3,\theta_4} ((mediaType = \text{"painting"} \rightarrow dominantColor \approx \text{RGB(...)}...$$

Im vorgestellten Verfahren werden die Gewichte nach einem logischen Transformationsschritt (siehe Abschnitt 3.1) während der arithmetischen Evaluierung ausgewertet [Zellhöfer and Schmitt, 2009]. Diese Integration ermöglicht die flexible Adaption der Gewichtung an die subjektive Wahrnehmung des Nutzers ohne die Semantik

[4] Sonderfall Evaluierung: Wenn eine Anfrage wie $q := x \approx A \vee x \approx B$ gegeben ist, dann gilt $q \equiv \exists t : (t = A \vee t = B) \wedge x \approx t$, d.h., erst die Anwendung von Transformationsregeln der Logik 1. Ordnung ermöglicht die Übersetzung der Anfrage in eine Form, die direkt von CQQL ausgewertet werden kann.

[5] Hier muss angemerkt werden, dass die Ölfarben der Renaissance charakteristisch altern, so dass Fachleute den kulturellen Ursprung eines Gemäldes schätzen können.

der Anfrage zu verletzen. Dieses flexible Vorgehen ist bei zuvor festgelegten Gewichten, wie z.B. während der oben genannten Aggregation verschiedener Repräsentationen, nicht ohne weiteres möglich.

Wie bereits erwähnt, stellt das Setzen der Gewichte eine komplexe Aufgabe für den Nutzer dar. Für eine Eingabe benötigt der Nutzer dabei Wissen um die Anfragesprache und Klarheit über seine subjektiven Präferenzen zwischen allen Repräsentationen, was die kognitive Belastung erhöht. Um eine einfache Bedienung zu gewährleisten, wird der Nutzer im vorgestellten Anfragemodell von einem RF-Prozess unterstützt, der auf maschinelles Lernen zurückgreift.

Nutzerinteraktion

Die Kernidee der Nutzerinteraktion basiert auf der Verwendung qualitativer Präferenzurteile als Eingabeverfahren für den Nutzer. Nach einer initialen Anfrage der Dokumentenkollektion werden potentiell relevante Dokumente dargestellt. Ist der Nutzer mit dieser Ergebnismenge unzufrieden, so kann die Anfrage und somit die Ausprägung der PCO iterativ angepasst werden. Dabei wird auf *induktive Präferenzen* [Zellhöfer, 2010b] auf den Ergebnisdokumenten zurückgegriffen. Induktive Präferenzen ermöglichen es dem Nutzer eine Präferenz zwischen zwei oder mehr Dokumenten wie Dokument d_1 ist relevanter als Dokument d_2 anzugeben. Abbildung 2 illustriert dies am Beispiel von Dokument $\#4$ ist besser als $\#3$. Das besondere an diesem Vorgehen ist, dass diese Beurteilung ohne Kenntnis der zugrundeliegenden Repräsentationen oder der Anfragesprache erfolgen kann. Vergleichbare Urteile sind aus dem Alltag bekannt, in dem häufig zwischen zwei Objekten gewählt wird, ohne dass alle ihre Eigenschaften detailliert bekannt sind. Folglich erhöht das vorgestellte Verfahren nicht die kognitive Last des Nutzers, was nicht bei allen RF-Verfahren der Fall ist [White, 2006].

Nach jeder Nutzerinteraktion müssen die Präferenzen einen gerichteten, azyklischen Graphen (ein Hasse-Diagramm) bilden, welcher als Eingabe für den maschinellen Lernalgorithmus dient, der die Gewichte der zugrundeliegenden CQQL-Anfrage entsprechend der Anforderungen des Nutzers anpasst. Es handelt sich dabei um einen angepassten Downhill-Simplex-Algorithmus [Nelder and Mead, 1965], der zur Lösung nichtlinearer Optimierungsprobleme geeignet ist. Aufgrund der arithmetischen Auswertung von CQQL (siehe Abschnitt 3.1) fällt das Lernverfahren in diese Klasse. Die angegebenen Präferenzen dienen dabei als Constraint. Eine detaillierte Beschreibung der zugrundeliegenden Algorithmen findet sich in [Zellhöfer and Schmitt, 2009; Schmitt and Zellhöfer, 2009] und wird hier aufgrund des anderen thematischen Fokus' ausgespart. Das Verfahren sammelt folglich im Laufe des Verfahrens eine Menge an Präferenzen an, die das Suchbedürfnis des Nutzers charakterisieren. In diesem Punkt ähnelt es dem *ostensive model* [Campbell, 2000], wobei jedoch mit der Zeit Präferenzen nicht durch einen Einflussfaktor „altern", sondern durch weitere ergänzt oder ersetzt werden.

Im Falle von unerfüllbaren Präferenzen werden diese durch den Algorithmus kommuniziert. Ein Beispiel für eine unerfüllbare Präferenz ist ein Zyklus wie: Dokument $A > B > C$ aber $C > A$. Solche Konflikte können einfach durch ein topologisches Sortieren bei jeder Modifikation des Präferenzgraphens erkannt werden. Je nach Art des Konflikts kann eine automatische Auflösung durch Priorisierung oder eine Pareto-Komposition angewendet werden [Zellhöfer, 2010a]. Durch die iterative Überprüfung des Graphen ist ebenfalls eine manuelle Korrektur durch den Nutzer möglich. Hierbei wird der Nutzer vom System dadurch unterstützt, dass es betroffene Präferenzen präsentiert und so eine klare Fallentscheidung ermöglicht. Letztendlich führen die geschilderten Lernschritte zu einer Adaption der PCO an die individuellen Vorstellungen des Nutzers. Dabei basiert die PCO nach wie vor auf einer hochstrukturierten Anfrage. Um die Retrievalqualität zu steigern ist es denkbar, dass Anfragen durch Domänenexperten vorformuliert oder empirisch bestimmt werden. Die Gewichte innerhalb der Anfragen steuern dann sozusagen den „Trend" innerhalb der Anfrageergebnisse, um diese weiter zu personalisieren.

4 Experimente und Nutzungsszenarien

Trotz des theoretischen Schwerpunkts der Arbeit wurde ein erstes Experiment durchgeführt, um den Nutzen des polyrepräsentativen Anfragemodells zu bewerten. Zusätzlich wurde das Verfahren in einer Real-Welt-Anwendung eingesetzt.

Das Experiment basiert auf einem Image-Retrieval-Szenario, welches auf 575 Urlaubsfotos aus Indonesien zurückgreift. Die PCO, d.h. die CQQL-Anfrage, wurde als gewichtete Konjunktion einiger LLF, die durch die LIRe-Bibliothek [Lux and Chatzichristofis, 2008] extrahiert wurden, modelliert. Um eine große Anzahl von Repräsentationen eines Dokuments bereitzustellen wurden die folgenden LLF ausgewählt: *SCALABLECOLOR, COLORLAYOUT, EDGEHISTOGRAM, AUTOCOLORCORRELOGRAM, COLORHISTOGRAM, GABOR, TAMURA, CEDD* und *FCTH*. Die ersten Features entstammen dabei dem MPEG-7-Descriptor-Set. Die letzen beiden sind aggregierte Features und werden in [Lux and Chatzichristofis, 2008] referenziert. Im aktuellen Experiment wurden noch keine auf HLF oder MD basierenden Repräsentation verwendet, da vor allem die Wirksamkeit der induktiven Präferenzen und des folgenden Lernalgorithmus' untersucht werden sollen.

Abbildung 5 (A) zeigt die gerankte Ergebnisliste basierend auf dem genannten PCO und einer initialen, neutralen Gewichtung[6]. Die erste Iteration nutzt eine explizit angegebene Präferenz als Eingabe für den Lernalgorithmus: das Dokument $\#5$ („Tempel") wird gegenüber dem Dokument $\#4$ („Hund") bevorzugt. Abbildung 5 (B) zeigt die darauf folgenden gerankten Ergebnisdokumente. Hier wird deutlich, dass bereits mehr Tempel in den oberen Resultaten erscheinen. Dies wird auch durch den Rankfehler deutlich. Der *Rankfehler* (Tabelle 2 [*Verfahren 2*]) zählt Veränderungen an den Rangpositionen der Dokumente zwischen zwei Rankings, womit sich der Einfluss der Präferenzen darstellen lässt. Abbildung 4 (unten) zeigt den Rankfehler zwischen dem initialen Rank und dem der ersten Iteration. Die schwarze Diagonale zeigt die Rankposition eines Dokuments nach der ersten Iteration während die roten Kreuze die ursprüngliche Position dieses Dokuments wiedergeben. D.h., dass das Dokument, welches nach der Präferenzmodifikation auf Position $\#3$ gerankt ist (3 auf der x-Achse), davor auf Rank $\#8$ war (y-Achse). Abbildung 5 (C) zeigt den neuen Rank nach der Hinzunahme zweier weiterer Präferenzen.

Alternativ kann ein anderes Verfahren der Präferenzerhebung Verwendung finden (Tabelle 2 [*Verfahren 1*]). In diesem Verfahren werden nicht nur explizit angegebene Präferenzen, wie eingangs geschildert, verwendet, sondern das Ranking der Ergebnisdokumente

[6]Es ist nicht notwendig mit einer neutralen Gewichtung zu beginnen. Beispiele für andere initiale Gewichtungen, wie die Verwendung von Nutzerprofilen, finden sich in [Zellhöfer and Schmitt, 2008].

Abbildung 2: Erhebung einer Präferenz zwischen Dokument 3 und 4. Dokument 4 wird bevorzugt.

Tabelle 2: Kennzahlen der beiden Verfahren zur Präferenzableitung

	Verfahren 1	**Verfahren 2**
Rankfehler (top-30)	26	23
Rankdistanz	0,08888889	0,11555555
Gewichtsdistanz	5,1999	6.639
Spearmans ρ	0,909	0,843

ausgenutzt. Bei diesem Verfahren werden Präferenzen aus dem Ranking solange abgeleitet bis vom Nutzer eine Modifikation vorgenommen wird, d.h. für das konkrete Beispiel, dass neben $\#5 > \#6$ auch noch die Präferenzen $\#1 > \#2$ und $\#2 > \#3$ hinzugenommen werden. Dieses Verfahren wirkt sich auch auf den neu generierten Rank aus, wie der unterschiedliche Rankfehler in Tabelle 2 [*Verfahren 1*] zeigt. Da das reine Abzählen von Positionsveränderungen noch nicht sehr aussagekräftig ist, stellt Tabelle 2 weitere Kennzahlen dar.

Die *Rankdistanz* $\delta_{rank}(r_x, r_y)$ (Gleichung 1; vgl. [Zellhöfer and Schmitt, 2009]) drückt die durchschnittliche Positionsdifferenz der Dokumente in zwei Ranks aus und kann damit aussagekräftiger als der Rankfehler ausdrücken, um wieviel Positionen Dokumente im Rank verschoben worden.

$$\delta_{rank}(r_x, r_y) = \frac{\sum_{i=1}^{n} |pos_{r_x}(d_i) - pos_{r_y}(d_i)|}{n^2}. \quad (1)$$

Wobei r_x und r_y zwei unterschiedliche Ranks sind, $pos_{r_x}(d_i)$ die Position von Dokument d_i in r_x und $pos_{r_y}(d_i)$ die Position in r_y ist. Beide Ranks bestehen dabei aus jeweils n Positionen.

Die *Euklidische Gewichtsdistanz* misst vor allem die Auswirkung angegebener Präferenzen auf die gelernten Gewichtswerte:

$$\delta_{weights}(w_1, w_2) = \sqrt{\sum_i |w_2(i) - w_1(i)|^2} \quad (2)$$

Hierbei sind w_1 und w_2 die gelernten Gewichtsschemata auf Grundlage verschiedener Präferenzen.

Außerdem wurde der Rangkorrelationskoeffizient *Spearmans* ρ berechnet, der in einem Intervall von $[-1; 1]$ liegt. Der Wert 1 steht dabei für identische Ranks und -1 für maximal unterschiedliche, d.h. umgedrehte Ranks. Im vorgestellten Experiment misst Spearmans ρ vor allem die Stärke der Veränderung der Ranks nach einer Iteration.

Stellt man beide Verfahren gegenüber, so wird deutlich, dass die alleinige Verwendung explizit angegebener Präferenzen (Verfahren 2) zu einer deutlicheren Veränderung eines Ranks nach einem Lernschritt führt. Ob sich dieser Unterschied auf die Erwartungskonformität des Systems auswirkt muss in Nutzerstudien untersucht werden. Zusammenfassend kann gesagt werden, dass das durchgeführte Experiment eine Wirksamkeit des vorgeschlagenen, polyrepräsentativen Anfragemodells aufzeigt und weitere, tiefergehende Studien motiviert.

In Ergänzung des durchgeführten Experiments wurde das vorgestellte Verfahren im Music-Retrieval-Projekt „GlobalMusic2one“[7] eingesetzt. Ziel dieses Projekts ist die Entwicklung eines Retrieval-Systems für Weltmusik, welches LLF, Tags, Folksonomien und regelbasierte Klassifikationen von Musik-Genres vereint. Die Klassifikationsregeln wurden dabei von Musikwissenschaftlern der HU Berlin bereit gestellt. Abbildung 3 zeigt eine solche Klassifikationsregel.

Im Rahmen eines Tests wurde eine PCO auf Grundlage dieser Regeln, die zum Teil auf LLF und relationalen Metadaten (Songlänge, Interpret o.ä.) basieren, modelliert, um die Ähnlichkeitssuche von Songs bzw. die Klassifikation dieser zu unterstützen. Subjektive Einschätzungen von Domänenexperten zeigen dabei, dass die Verwendung des vorgeschlagenen Anfragemodells der alleinigen Verwendung von LLF überlegen ist. Es wird vermutet, dass dies an der strukturellen Mächtigkeit von CQQL liegt, welche es ermöglicht, polyrepräsentative Konzepte, die der Musik inhärent sind, korrekt widerzuspiegeln.

Obwohl die ersten Untersuchungen bereits motivierende Ergebnisse liefern, sind weitere Experimente notwendig, um belastbare Erkenntnisse zu gewinnen. Offenbar ist die strukturierte Kombination von Repräsentation aus verschiedenen Retrieval-Domänen möglich. Wobei die Einbeziehung von Gewichten vor allem der Personalisierung der Ergebnismenge dient, welche letztendlich die Nutzerzufriedenheit erhöht.

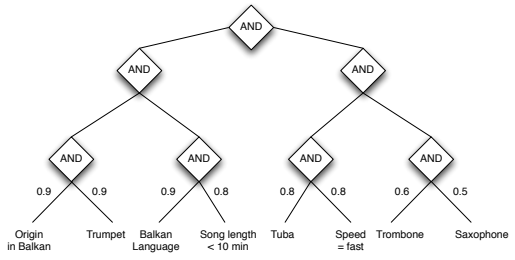


Abbildung 3: Auszug aus dem CQQL-Bedingungsbaum für das Genre „gypsy brass music“. (Die Gewichte für die Konjunktionen sind an den Kanten angegeben.)

5 Fazit und Ausblick

Das vorliegende Paper stellt ein kognitiv motiviertes Anfragemodell für das Multimedia Retrieval vor. Das vorge-

[7] http://www.globalmusic2one.net/

schlagene Modell nutzt CQQL, um das polyrepräsentative Prinzip [Ingwersen and Järvelin, 2005] mittels einer hochstrukturierten Anfragesprache umzusetzen. Zeitgleich wird der Nutzer durch ein RF-Verfahren, welches Methoden des maschinellen Lernens verwendet, bei der Personalisierung der entstandenen Cognitve Overlap aller zur Verfügung stehenden Repräsentationen eines Dokuments unterstützt. Das gesamte Verfahren stellt dabei ein theoretisch begründetes Framework für Polyrepräsentation dar, welches leicht auf andere Retrieval-Bereiche ausgedehnt werden kann.

Das angesprochene maschinelle Lernverfahren wird zur Personalisierung der Cognitive Overlap eingesetzt, um die Nutzerzufriedenheit zu erhöhen. Zur Gewährleistung einer niedrigen Belastung für den Nutzer greift das RF auf induktive Präferenzen zurück, mit denen es möglich ist, „besser-als"-Beziehungen auf den Ergebnisdokumenten anzugeben. Hierbei bleibt das zugrundeliegende Anfragemodell für den Nutzer verborgen. So kann eine intuitive Bedienung sichergestellt werden.

Im folgenden sollen einige offene theoretische und praktische Fragen vorgestellt werden. Durch die Umsetzung des polyrepräsentativen Prinzips erscheint es sinnvoll das diskutierte Framework in das „polyrepresentation continuum" [Larsen *et al.*, 2006] einzuordnen. Neben dieser theoretischen Klassifizierung steht die Entwicklung einer graphischen Oberfläche für das Anfragemodell zur Durchführung von Nutzerstudien im Mittelpunkt. Dabei ist geplant das polyrepräsentative Prinzip auch in der GUI widerzuspiegeln. Diese Idee wird durch [White, 2006] unterstützt, welcher das Prinzip erfolgreich zur Dokumentenvisualisierung und Navigation einsetzt. Die Visualisierung betrachten wir dabei als besondere Herausforderung, da sie die polyrepräsentativen Charakteristika eines Dokuments wiedergeben muss. Hierfür reicht die simple Darstellung von Thumbnails oder Video-Zusammenfassungen nicht mehr aus.

Auf der anderen Seite sind weitere Experimente nötig, um die Retrieval-Qualität des diskutierten Ansatzes zu bewerten. Aufgrund des starken Einflusses an Subjektivität durch die Verwendung von induktiven Präferenzen bieten sich hier die traditionellen Maße wie Precision und Recall nur beschränkt an. Vielmehr ist der Einsatz des *Normalized Discounted Cumulative Gains* (NDCG) [Järvelin and Kekäläinen, 2002] sinnvoll, da es gestufte Relevanzbeurteilungen zulässt, mit denen bei dem vorliegen Anfragemodell zu rechnen ist. Als Ausgangsbasis dient die bereits erwähnte Photosammlung mit 575 Dokumenten, die bereits mittels abgestufter Relevanzurteile durch verschiedene Nutzer bewertet wird. Im Anschluss an diese Untersuchungen sind erste Usability-Tests der GUI denkbar.

Mittelfristig muss die Modifikation der PCO untersucht werden. Dies wird notwendig, da damit zu rechnen ist, dass sich das Suchbedürfnis der Nutzer verschieben kann und so nicht mehr durch die initiale PCO abgedeckt wird. Hierbei kann das diskutierte Lernverfahren modifiziert werden, so dass konkrete CQQL-Anfragen gelernt werden können, die aus angegebenen Präferenzen abgeleitet werden.

Literatur

[Aucouturier and Pachet, 2004] J J. Aucouturier and F. Pachet. Improving Timbre Similarity: How high is the sky? In *Journal of Negative Results in Speech and Audio Sciences*, volume 1 of *1*. 2004.

[Birkhoff and Neumann, 1936] G. Birkhoff and J. Neumann. The Logic of Quantum Mechanics. *Annals of Mathematics*, 37:823–843, 1936.

[Campbell, 2000] Iain Campbell. Interactive Evaluation of the Ostensive Model: Using a New Test Collection of Images with Multiple Relevance Assessments. *Inf. Retr.*, 2(1):89–114, 2000.

[Fagin and Wimmers, 2000] R. Fagin and L. E. Wimmers. A Formula for Incorporating Weights into Scoring Rules. *Special Issue of Theoretical Computer Science*, (239):309–338, 2000.

[Frommholz and van Rijsbergen, 2009] I. Frommholz and C.J. van Rijsbergen. Towards a Geometrical Model for Polyrepresentation of Information Objects. In *Proc. of the "Information Retrieval 2009" Workshop at LWA 2009*, 2009.

[Hull, 1997] A. David Hull. Using Structured Queries for Disambiguation in Cross-Language Information Retrieval. In *AAAI Spring Symposium on Cross-Language Text and Speech Retrieval Electronic Working Notes*, pages 24–26, 1997.

[Ingwersen and Järvelin, 2005] Peter Ingwersen and Kalervo Järvelin. *The Turn: Integration of Information Seeking and Retrieval in Context*. Springer-11645 /Dig. Serial]. Springer, Dordrecht, 2005.

[Järvelin and Kekäläinen, 2002] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446, 2002.

[Larsen *et al.*, 2006] Birger Larsen, Peter Ingwersen, and Jaana Kekäläinen. The polyrepresentation continuum in IR. In *IIiX: Proceedings of the 1st international conference on Information interaction in context*, pages 88–96. ACM, 2006.

[Lee *et al.*, 1993] H. J. Lee, Y. W. Kim, H. M. Kim, and J. Y. Lee. On the Evaluation of Boolean Operators in the Extended Boolean Retrieval Framework. In Robert Korfhage, Edie M. Rasmussen, and Peter Willett, editors, *ACM/SIGIR 1993, Proceedings of 16th Annual International Conference on Research and Development in Information Retrieval, Pittsburgh, USA*, pages 291–297, 1993.

[Lux and Chatzichristofis, 2008] Mathias Lux and A. Savvas Chatzichristofis. Lire: Lucene Image Retrieval: An Extensible Java CBIR Library. In *MM '08: Proceeding of the 16th ACM International Conference on Multimedia*, pages 1085–1088. ACM, 2008.

[Nelder and Mead, 1965] A. J. Nelder and R. Mead. A Simplex Method for Function Minimization. *Computer Journal*, 7:308–313, 1965.

[Rocchio, 1971] J. Rocchio. Relevance Feedback in Information Retrieval. In *The SMART Retrieval System*, pages 313–323. 1971.

[Salton *et al.*, 1983] Gerard Salton, A. Edward Fox, and Harry Wu. Extended Boolean Information Retrieval. *Commun. ACM*, 26(11):1022–1036, 1983.

[Schmitt and Zellhöfer, 2009] Ingo Schmitt and David Zellhöfer. Lernen nutzerspezifischer Gewichte innerhalb einer logikbasierten Anfragesprache. In Christoph Johann Freytag, Thomas Ruf, Wolfgang Lehner, and Gottfried Vossen, editors, *Datenbanksysteme in Business, Technologie und Web (BTW 2009), 13. Fachtagung des GI-Fachbereichs "Datenbanken und Informationssysteme (DBIS), Proceedings, 2.-6. März 2009, Münster, Germany*, volume 144 of *lni*, pages 137–156. GI, 2009.

[Schmitt, 2008] Ingo Schmitt. QQL: A DB&IR Query Language. *The VLDB Journal*, 17(1):39–56, 2008.

[Skov *et al.*, 2004] Metter Skov, Henriette Pedersen, Birger Larsen, and Peter Ingwersen. Testing the Principle of Polyrepresentation. In Peter Ingwersen, C.J. van Rijsbergen, and Nick Belkin, editors, *Proceedings of ACM SIGIR 2004 Workshop on "Information Retrieval in Context"*, pages 47–49, 2004.

[Turtle and Croft, 1991] Howard Turtle and Bruce W. Croft. Evaluation of an inference network-based retrieval model. *ACM Trans. Inf. Syst.*, 9(3):187–222, 1991.

[van Rijsbergen, 2004] C.J. van Rijsbergen. *The Geometry of Information Retrieval*. Cambridge University Press, Cambridge, England, 2004.

[White, 2006] W. Ryen White. Using searcher simulations to redesign a polyrepresentative implicit feedback interface. *Inf. Process. Manage.*, 42(5):1185–1202, 2006.

[Zellhöfer and Schmitt, 2008] David Zellhöfer and Ingo Schmitt. A Poset Based Approach for Condition Weighting. In *6th International Workshop on Adaptive Multimedia Retrieval*. 2008.

[Zellhöfer and Schmitt, 2009] David Zellhöfer and Ingo Schmitt. A Preference-based Approach for Interactive Weight Learning: Learning Weights within a Logic-Based Query Language. *Distributed and Parallel Databases*, 2009.

[Zellhöfer, 2010a] David Zellhöfer. Eliciting Inductive User Preferences for Multimedia Information Retrieval. In Wolf-Tilo Balke and Christoph Lofi, editors, *Proceedings of the 22nd Workshop "Grundlagen von Datenbanken 2010"*, volume 581, 2010.

[Zellhöfer, 2010b] David Zellhöfer. Inductive User Preference Manipulation for Multimedia Retrieval. In Laszlo Böszörmenyi, Dumitru Burdescu, Philip Davies, and David Newell, editors, *Proc. of the Second International Conference on Advances in Multimedia*, 2010.

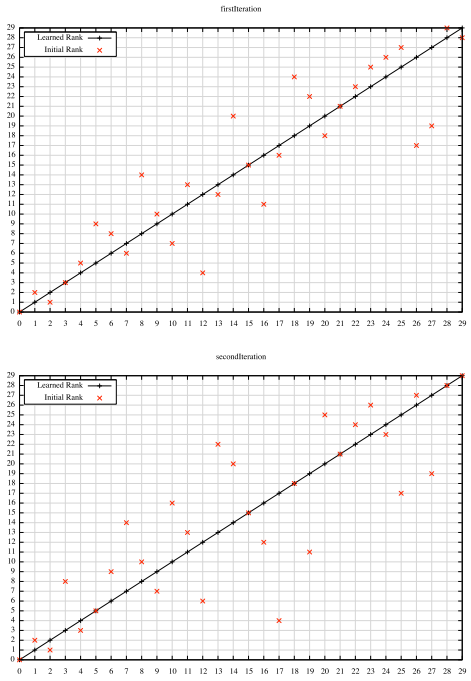


Abbildung 4: Rankfehler für Verfahren 1 (oben) und Verfahren 2 (unten) basierend auf der Präferenz aus der ersten Iteration aus Abbildung 5.

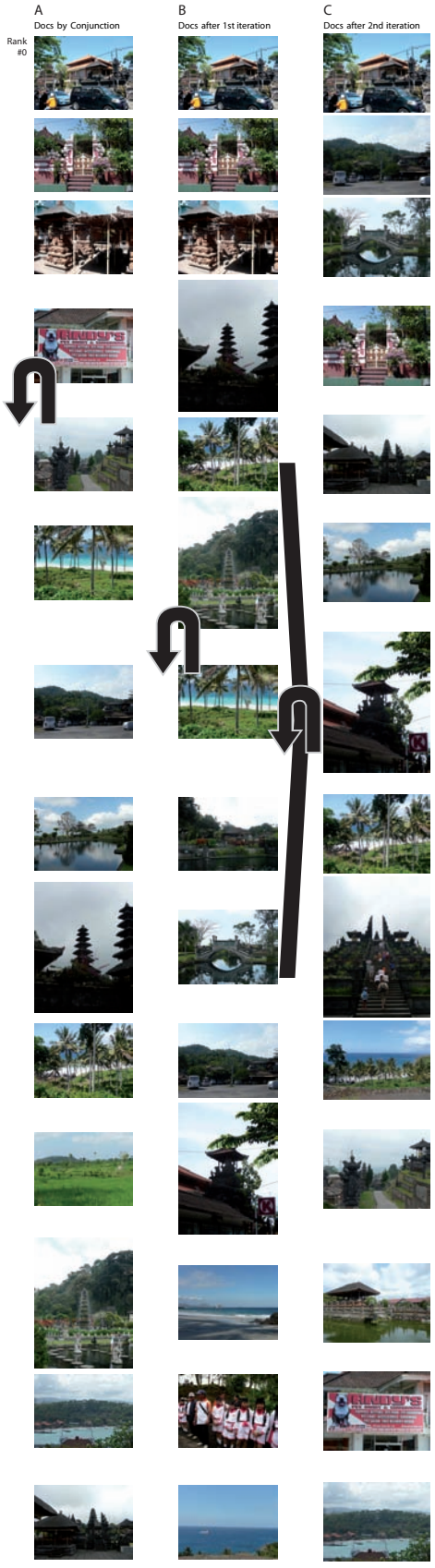


Abbildung 5: Ausschnitt aus der Ergebnisliste nach zwei Iterationen. (Die Pfeile geben umgedrehte Präferenzen zwischen zwei Dokumenten an.)

Workshop on Knowledge and Experience Management FGWM 2010

Joachim Baumeister
Intelligent Systems
Computer Science VI
University of Würzburg
joba@uni-wuerzburg.de

Thomas Roth-Berghofer
University of Hildesheim
& German Research Center for
Artificial Intelligence DFKI GmbH
thomas.roth-berghofer@dfki.de

The FGWM Workshop Series

The workshop "Knowledge and Experience Management" is organised annually by the Special Interest Group Knowledge Management (Fachgruppe Wissensmanagement, in short FGWM) of the German Informatics society (GI). The main goal is to enable and further the exchange of innovative ideas and practical applications in the field of knowledge and experience management. Also, it aims to provide an interdisciplinary forum.

Submissions from current research out of these and adjacent areas were welcome. Moreover, contributions that describe work in progress or approaches that have not yet been investigated comprehensively were of special interest. The workshop provided the opportunity for young researchers to present and discuss preliminary work, get feedback on their work from a larger audience with people having quite different viewpoints, get accustomed to presenting and defending their work at scientific events. As every year, the workshop was a forum for testing the viability of new ideas, by junior as well as senior researchers, before more effort and resources are put into them.

FGWM 2010

The proceedings contain the papers presented at FGWM 2010 held on October 4–6, 2010 in Kassel, Germany. Six research papers and one resubmission of previously published work were accepted. For the latter only a one page abstract is included in the proceedings. The topics of interest of the FGWM workshop series are:

- Experience/knowledge search and knowledge integration approaches (case-based reasoning, logic-based approaches, text-based approaches, semantic portals/wikis/blogs, Web 2.0, etc.)
- Applications of knowledge and experience management (corporate memories, e-commerce, design, tutoring/e-learning, e-government, software engineering, robotics, medicine, etc.)
- (Semantic) Web Services for knowledge management
- Agile approaches within the KM domain
- Agent-based & Peer-to-Peer knowledge management
- Just-in-time retrieval and knowledge capturing
- Ways of knowledge representation (ontologies, similarity, retrieval, adaptive knowledge, etc.)
- Support of authoring and maintenance processes (change management, requirements tracing, (distributed) version control, etc.)
- Evaluation of knowledge management systems
- Practical experiences ("lessons learned") with IT aided approaches
- Integration of knowledge management and business processes
- Introspection and explanation capabilities of knowledge management systems
- Linked (Open) Data

Programme Comittee

- Andreas Abecker, FZI Karlsruhe
- Klaus-Dieter Althoff, University of Hildesheim
- Joachim Baumeister, University of Würzburg
- Ralph Bergmann, University of Trier
- Andrea Kohlhase, Jacobs University Bremen
- Ronald Maier, University of Innsbruck, Austria
- Heiko Maus, DFKI GmbH
- Mirjam Minor, University of Trier
- Markus Nick, empolis GmH
- Ulrich Reimer, University of Applied Sciences St. Gallen, Switzerland
- Bodo Rieger, University of Osnabrück
- Thomas Roth-Berghofer, University of Hildesheim & DFKI GmbH
- Rainer Schmidt, University of Rostock
- Steffen Staab, University of Koblenz-Landau

Additional Reviewers Daniel Bachlechner, Christopher Harb, and Leo Sauermann

We thank the Program Committee and, particularly, the additional reviewers for their efforts and good work.

If you would like to participate in further discussions or receive further information about future workshops you might consider joining the FGWM mailing list [1].

August 2010 — Joachim Baumeister, Thomas Roth-Berghofer

Acknowledgements This volume has been produced in part using the EasyChair system[2]. We would like to express our gratitude to its author Andrei Voronkov.

[1] http://www.fgwm.de/newsletter.html
[2] http://www.easychair.org

Knowledge System Prototyping for Usability Engineering

Martina Freiberg, Johannes Mitlmeier, Joachim Baumeister, Frank Puppe
University of Würzburg
D-97074, Würzburg, Germany
freiberg/joba/puppe@informatik.uni-wuerzburg.de

Abstract

Knowledge-based consultation and documentation systems are widely distributed in industrial and medical environments today. Yet, their implementation still is a tedious and costly task. Furthermore, the aspect of usability—which is in principle of critical importance for those systems—is often nearly unconsidered. We argue, that tailored UI prototyping can help to tackle both issues. Therefore, we propose a UI prototyping tool for knowledge-based systems, intended to enhance knowledge systems engineering in itself, and to foster usability engineering in that context.

Keywords: Knowledge-based System, User Interface Prototyping, Usability, Human Computer Interaction

1 Introduction

Knowledge systems (KS)—in the context of our paper knowledge-based consultation and documentation systems—are applied in various industrial and medical environments today. Regarding their development, it has to be differentiated between knowledge system engineering (KS Engineering) and knowledge engineering (KE). The former comprises the entire development process of a knowledge-based system, including especially its UI and interaction design; the latter specifically addresses the definition and formalization of the required knowledge, e.g., the terminology, or explicit problem-solving knowledge.

KS Engineering Pitfalls and SE Solutions

Despite the widespread use of knowledge systems, and consequently increasing research efforts regarding their development in the last decades, KS Engineering still remains a tedious and complex task. Among the main pitfalls are high development costs, both in terms of money and time. Thereby, the sub-task KE alone often causes a major part of the expenses, hence influencing other KS Engineering activities:

In many cases, UI- and interaction design in general—or a more targeted comparison of several equal design options—would require more attention. For knowledge based systems, it is further of critical importance, that they are intuitive and easy to use as to not distract the users from the often difficult, domain-specific jobs, they are intended to support (e.g., decision-support in medical contexts). Yet, despite various recognized general usability engineering approaches, usability issues often also remain almost unconsidered. Finally, the often still missing true understanding of such systems and their benefits on the side of potential customers, can make it difficult to promote respective projects in the first place due to the overall complexity and costs.

In general software engineering (SE) and in human-computer interaction (HCI), user interface (UI) prototyping is an established method for iterative specification and refinement before implementing the productive system [3; 4]. The increased flexibility arising from prototyping-based specification and design offers the chance to adapt system (and especially interface) requirements to changing base requirements or customer wishes in a more efficient, inexpensive manner. The affordability of the approach also permits the early evaluation and comparison of design alternatives. In providing a (visual) basis for communication, UI prototyping can help to specify requirements more precisely. Thus, the potential risks of fundamental misunderstandings, and a resulting, more expensive redesign of central conceptions at a later stage of the project, can be reduced. Also, the overall system vision can be communicated and refined more easily with the help of an appropriate prototype.

In tailoring UI prototyping for knowledge systems, we aim at exploiting those advantages. Particularly, we intend to foster affordable, pragmatic KS Engineering, that both helps to promote respective projects in the first place, and alleviates the overall task. Our approach intentionally focusses on interface and interaction design of KS, and on an increased integration of usability considerations in the process. Regarding the specification and formalization of the required terminology and explicit knowledge of a KS, there exist various established KE methods today, each of which can be equally well applied—thus, we do not further discuss or value their suitability here.

Related Work

In general SE and in HCI there exist numerous classifying approaches regarding prototyping and the prototyping process, e.g., see [3; 4; 5; 8; 11]. Apart from manifold general prototyping tools and methodologies available—see Beaudouin-Lafon and Mackay [4] for an overview—also tailored tools have been developed in various specific domains; examples are the field of multimodal interaction research [12], or the field of cross device interface design [10]. To the best of our knowledge, no prototyping approaches or tools exist, that specifically address knowledge systems according to our definition (see Section 3.1).

Lim et al. [9] note that in HCI, prototypes to date

mainly are used for evaluation purposes—such as usability testing—and that in SE, prototyping mostly constitutes a means for supporting requirements engineering; in extension to that, they suggest prototypes as tools for designers to frame, refine, and discover options in a design space. Regarding an integration of software- and usability engineering, Memmel et al. [13] similarly claim an incorporation of visual requirements engineering—based on appropriate prototypes—much earlier in the overall process. Merging those insights, we propose UI prototyping as a rather pragmatic means for KS requirements engineering. As opposed to the above approaches, we explicitly address knowledge systems, that exhibit some specific characteristics. First, KS mostly consist of a rather fixed set of UI elements and user-system interactions; most often, for example, questions are presented, answered by the user, and cause a certain follow-up system (re)action. An adequate prototyping tool thus firstly should support the design of such elementary elements as flexibly as possible. For realistically emulating actual knowledge systems, additionally the imitation—or actual integration—of the underlying explicit knowledge needs to be supported, as to enable a reasonable judgement regarding the applicability and usability of the overall future system.

When designing with usability in mind, some kind of iterative process is highly advisable [14]. Angele et al. [1] introduce a cyclic process model for developing knowledge-based systems; it incorporates prototyping techniques, but furthermore also formal specification and KE activities, thus constituting an entire, rather heavyweight engineering approach. Contrastingly, we suggest an extension of the rather lightweight *Agile Process Model* [2]; thereby, the focus is on providing an overall pragmatic method of KS Engineering and on enabling a rather inexpensive integration of usability activities.

In summary, we contribute to current research by

- proposing an overall approach that integrates efficient, affordable KS Engineering and usability engineering.
- introducing the UI prototyping and engineering tool *ProET*, specifically tailored for the design of knowledge-based systems

The remainder of the paper is organized as follows: In Section 2, we discuss a customized, prototyping-based KS Engineering process, as well as its potential regarding an integration of usability activities. We present *ProET*, an UI prototyping and engineering tool for knowledge-based systems, in Section 3. In Section 4, we report on experiences of exemplarily recreating existing knowledge systems with the tool, and on its consequential current scope and limitations. We conclude with a summary and an outlook to further research directions in Section 5.

2 Pragmatic KS Engineering for Usability

Regarding knowledge system development and knowledge engineering, there exist diverse approaches today, such as *CommonKADS*, *MIKE*, or adaptions of the classical *stage-based* and *incremental* software development models. Yet, for the success of knowledge system projects also and especially regarding small to mid-sized companies, a pragmatic approach—affordable and efficient regarding time and effort—is essential, c.f. [2]. Especially for promoting such projects in the first place, it is important to quickly

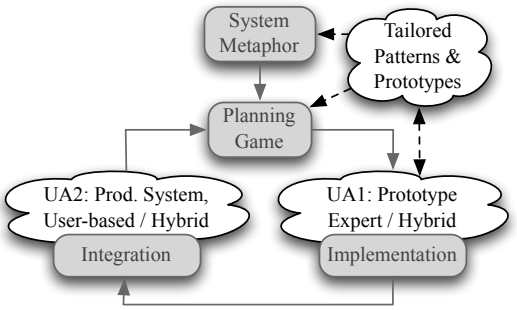


Figure 1: Extended Agile Process Model.

come up with first solutions, e.g., in the form of prototypes or example implementations. In this respect, we made positive experiences with applying the *Agile Process Model*, described in [2]. However, that model emphasizes knowledge base development, not yet taking much into account the design of the target system's interface, or usability traits.

Targeting an overall approach that supports pragmatic, affordable, and usability-involving KS Engineering, we propose the extension of the *Agile Process Model* by integrating pattern-based design, prototyping, and usability techniques into the original model. Figure 1 introduces the entire resulting *Extended Agile Process Model*. Although pattern integration and respective activities are included in the following for reasons of completeness, their more detailed discussion is part of further work, see [7].

The gray parts of Figure 1 represent the original model, consisting of the four phases *System Metaphor*, *Planning Game*, *Implementation*, and *Integration*. For a detailed discussion, see [2].

Basically, tailored patterns and prototyping can support both *System Metaphor* and *Planning Game*. In *System Metaphor*, the system objectives are defined by developers and customers. Based on appropriate patterns and corresponding implementation examples, a basic idea can be developed more easily. Thereby, patterns can be assessed either manually, or by using a tailored recommender system, that suggests patterns matching the target context. Prototypes, that also provide the relevant user-system interactions, further support that process by presenting a realistic simulation of a potentially resulting system as opposed to the static, visual depiction of knowledge system examples provided by the patterns.

The *Planning Game* defines the scope and prioritization of development tasks. Here, patterns ease the analysis and valuation of system requirements—taking place during the *Exploration* sub-phase of the planning game—by providing clear specifications of required features and interactions. Additionally, prototyping supports that task by allowing for actually trying out (and thus better evaluating) relevant functionalities.

With regards to *Usability Activities*, the original model can be extended both regarding *Implementation* and *Integration* (Figure 1, UA1, UA2). The basic model defines *Implementation* as a test-first activity—i.e., before actually implementing new or additional features, the corresponding tests for assuring their correctness are developed. This can be expanded by an evaluation-first activity, in the sense that based on the formerly created pro-

totypes, usability issues are assessed and valued first, before continuing with test-first implementation as defined by the model. Performing prototype-based usability evaluation offers the chance to reveal defects of the design at early stages. This can considerably lower development costs, as the adaption/revision of a prototype is rather inexpensive, in contrast to adapting an preliminarily implemented, or even already productive system. Without going into detail here, at that stage, expert- or hybrid approaches (according to a categorization suggested in [6]) seem to be most appropriate. For example, rather light-weight techniques—as feature-/consistency inspection—but also more comprehensive methods—as heuristic evaluation or expert walkthrough—are performed by the developer ("expert"). In case even future users—e.g., project partners or their employees—are available, hybrid methods such as pluralistic walkthrough or participatory/cooperative heuristic evaluation potentially can provide the most benefits. However, some of those techniques require at least a partly functional system—as explained in Section 3, also fundamental interactions of knowledge systems can be designed/simulated with the suggested prototyping tool *ProET*. Thus, those techniques are (at least partly) applicable to the developed UI prototypes.

During *Integration*, the implemented functionality is added to the productive system, using integration tests for assuring its overall correctness and integrity. Such testing can be extended by usability evaluation activities that check, whether the system still meets the specified usability goals. As *Integration* results in a running version of the productive system, it is not only possible, but rather highly advisable, to evaluate the applicability of the system in the target context with real users. Thus, not only hybrid, but also purely user-based usability evaluation techniques are beneficial—example techniques are querying, user studies, or controlled experiments. Additionally, again also *Hybrid Approaches* may also provide valuable insights regarding the actual use of the knowledge system and potential, remaining defects.

The suggested approach aims at turning overall KS Engineering into a more pragmatic process, equally suitable for promoting KS projects—by quickly setting up and presenting actual KS examples (prototypes) to customers—and for specifying requirements as well as the system design in a more agile manner. Due to the highly iterative process, that also involves usability evaluation activities at specified stages, potential system flaws may be detected, or even prevented more effectively.

3 The Prototyping Tool *ProET*

In this section, we introduce the prototyping and engineering tool *ProET*, that we are developing to support affordable and efficient KS Engineering, as well as to foster an eased integration of usability-related activities in the overall process. Therefore, we first define the specific type of target systems, as well as typical components those systems are built of. Afterwards, we introduce *ProET* and its workflow of creating prototypes in more detail.

3.1 Target Knowledge Systems and Components

By *knowledge system*, we understand systems that may implement various forms of knowledge—such as rules, or covering models—to support the user as efficiently as possible in performing the task at hand. Thereby, we specifically think of consultation and documentation tasks—in the first case, the system provides decision-support or recommendations regarding a specified problem area (e.g., in medical or fault detection contexts); in the second case, users are assisted in entering a certain set of data and the system ensures its quality (e.g., measured by completeness and correctness). The initial capabilities of the tool are based on our past experiences with developing knowledge-based systems. For the greater part, those were implemented as *web-based systems*—not necessarily meaning they are made available to large masses of users via internet, but in the general sense of "running in a browser". Thus, the tool specifically supports web-based consultation and documentation systems engineering.

For those target systems, typically a certain set of *visible knowledge components* can be identified:
- *Questions:* Requesting required input data from the user
- *Questionnaires:* May be used to group the (potentially large set of) questions
- *Answers:* A fixed set of reasonable input data (answer alternatives) to choose from, or free text input facilities
- *Solutions:* Fault or medical diagnoses, or action recommendations, that are derived by the included diagnosis knowledge (*invisible knowledge components*, e.g., rules)
- *Ancillary Information:* Informal knowledge representations, detailed elaborations of questions/solutions, or add-on information regarding the overall consultation/documentation progress.

Those components form the conceptual basis of the widgets currently supported by *ProET*. Thus, they constitute the elementary items that are to be specified in the declarative prototype specification file (as described in the following section in more detail).

3.2 Introducing *ProET*

The **pro**totyping and **e**ngineering **t**ool *ProET* is an UI prototyping tool specifically tailored for web-based consultation and documentation systems. Thereby, prototyping is supported gradually: First, exemplary system definitions (and corresponding templates/styles) allow for quickly and easily creating and exploring the basic collection of knowledge systems supported so far. Based on those available specifications, adapted KS prototypes can be created in a copy & modify manner, where the degree of modification can vary arbitrarily. With the extensibility of the tool, finally also entirely different interfaces/UI components can be developed and integrated, if required (see Section "Extending *ProET*").

The tool supports two basic modes of prototyping: Complete specification of all textual elements, as well as their (partly or entire) auto-generation.

The first variant is useful for prototyping and evaluating concrete KS ideas. The required visible knowledge components (such as questions, answers...) are defined in the declarative prototype specification file. As also elementary interactions—e.g. coloring the next suggested question, or hiding/unfolding parts of the dialog—are available, future knowledge systems can be simulated rather realistically. Thus, it can be examined whether a chosen UI/interaction design is suitable in a given, domain-specific context.

The option of auto-generating textual elements, further-

more allows for a more design-oriented prototyping: Not having to consider the specification of actually reasonable knowledge base elements simplifies the concentration on UI/interaction design questions. This can be helpful in case several designs are to be compared against each other, or provided that general design issues need to be evaluated, independent from any future system.

Technical Basis

ProET is a JAVA application that integrates several web-based technologies for engineering UI prototypes of web-based knowledge systems. The created prototype is HTML-based, enriched by JavaScript/AJAX for interactivity and styled by CSS. The tool is probably most comfortably used from within some kind of IDE—such as Eclipse[1]—that supports editing of the required file formats, as well as an easy management of the project itself.

Prototyping Workflow

To provide a first impression, Figure 2 presents a questionary-style, partly auto-generated consultation system prototype. Figure 2 (A) displays a page containing concretely specified questions and answers; another page of the same prototype, with auto-generated textual elements, is shown in Figure 2 (B). Prototyping with ProET currently is purely text-based. We use the above prototype as a running example when introducing the three types of specification files required for prototyping with *ProET*:

- An *XML-based* specification file (central prototype- and textual content specification)
- *String Template* files (creating HTML-/JavaScript-based fragments for each defined component)
- *CSS* files (design definition of specified components)

Once the specification via these files is finished, the prototype is assembled: Filling in the textual contents from the XML specification, HTML-/JavaScript-based component representations are created using the String Templates for the framework and CSS for the concrete styling.

XML-based, Elementary Prototype Specification The elementary prototype specification—i.e., its skeletal structure, consisting of the textual elements as well as of their basic UI properties (e.g., whether the question-style is one-choice or multiple-choice)—is provided in an XML-based format. For each of the visible (textual) knowledge system components, matching tags are provided—e.g., an `<answer>` tag is used for defining answers. Within those tags, the fundamental UI properties are specified as attributes—a multiple choice question, for example, is defined using `<question answer-type='mc'>`. Furthermore, the XML specification references the respective template- and CSS files, that are additionally required for creating the prototype.

Figure 2 (C) presents an excerpt of the file used for specifying the prototype shown in (A). Excluding the standard XML-header, the framing `<dialog...>` tag is the topmost element; there, the knowledge system type—here: `type='gen'`, referring to a predefined, (partly) auto-generated prototype style—is defined, as well as references to the respective template namespace, and style files. Figure 2 (D) exemplifies the specification of a questionnaire (here still named `page`) that consists of several questions; due to space reasons, only the detailed definition of the first question is printed completely. The example illustrates the specification of the question `Do you like surveys?`, the setting of its basic UI style by `answer-type='oc'`, as well as of its three answer alternatives `Yes, No, Neither...Nor`. Furthermore, the definition of generated textual elements is exemplified in Figure 2 (E). This is achieved by using `<generate>` tags, that can be attributed by the desired number of questions of the generated page (`num-questions='3'`) and respective answer alternatives (`num-answers='3'`), by the text lengths (`question-/answer-length='...`) or by the basic answer-style (as above). Regarding the auto-generation, it is possible to either define the number of questions (answers, text/answer length) strictly—e.g. `num-questions='3'`—or to specify a more flexible range—e.g. `num-questions='2-5'`—resulting in randomly calculated minimum of two and maximum of five questions.

String Templates String Template files provide the HTML-counterparts for each of the defined knowledge system components. When creating a prototype, one framing template is defined for the dialog as a whole—Figure 2 (F)—which contains the skeletal HTML-framework. From there, sub-templates are referenced, that define the HTML fragment of the respective components separately—in the figure, `$children$` means that the also depicted template for the *children* of the *dialog-content* element—which are pages—is inserted at this point. Splitting the UI templates into framing- and sub-templates, according to the corresponding component definitions, provides the advantage that the separate component templates, e.g. for a page, can be reused in several different prototypes.

CSS for UI styling The actual styling/design of the components is finally specified using standard *CSS*. Most basically, each knowledge system component can be globally styled by a CSS class with a matching name—e.g., common properties of all questions can be set by a `.question` class. Yet, a more fine-granular styling is also possible—additional, element-specific CSS classes or IDs can be set within the template files, and then can also be specified separately in the CSS file.

Interactivity With *ProET* not only static UI designs can be created, but also elementary interactivity that would be expected of an actually implemented knowledge system. Examples are the interactive coloring of questions (or solutions) according to their current status—e.g., answered/suggested next (or established/excluded)—or hiding parts of the questions until another defined question was answered by the user. Such interactive behavior is achieved by the usage of JavaScript in *ProET*. The required functionality thereby is defined in separate JS files, whereas the corresponding function calls are inserted in the String Template file where needed—e.g., if a given element should provide some interactivity on mouse-click, in the template file like `onClick="javascript:doSomething()` is inserted within the respective tag defining that element.

[1] http://www.eclipse.org/

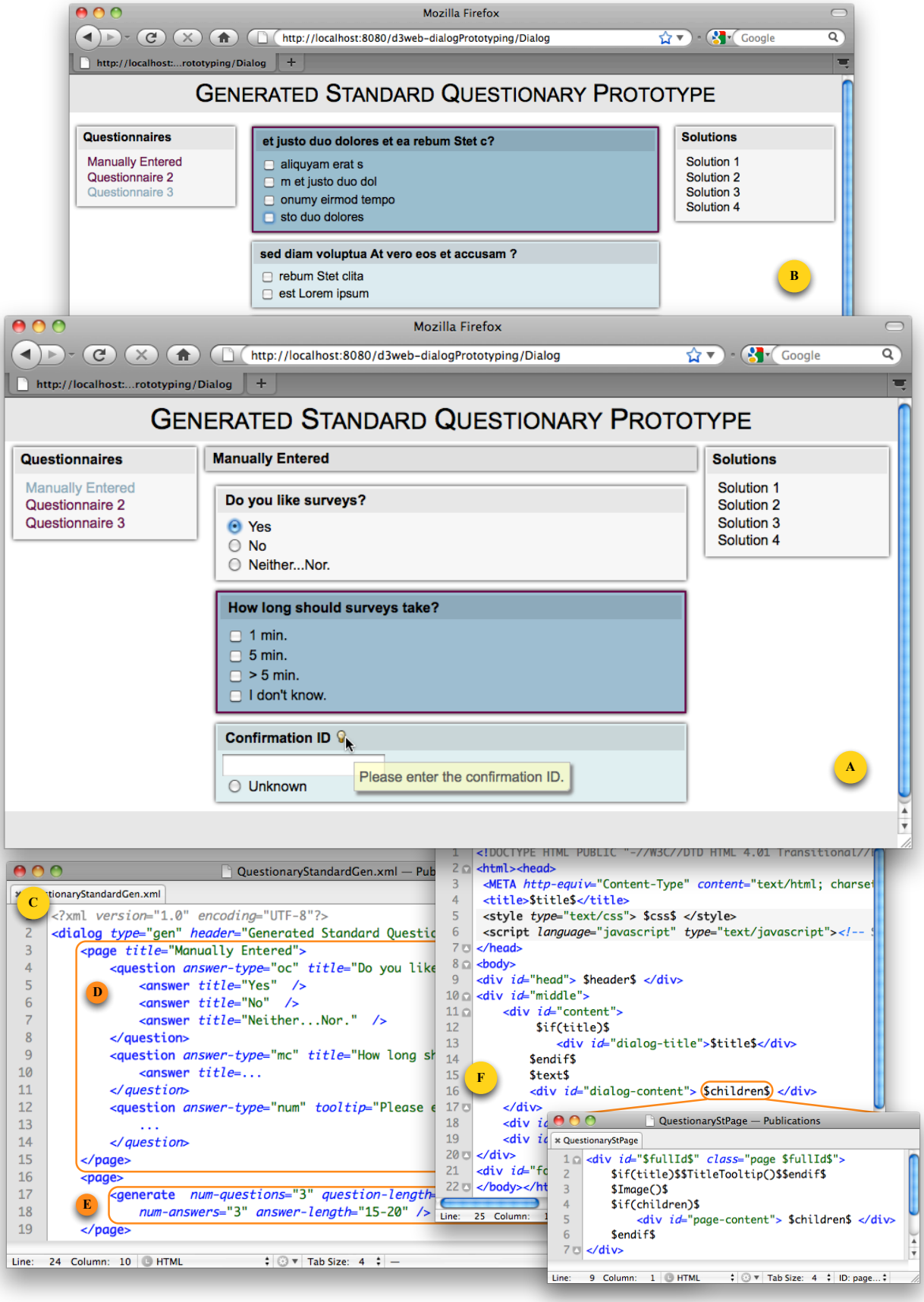


Figure 2: Partly auto-generated prototype of a questionary-like consultation system—manually entered questions (A), auto-generated questions (B), the corresponding prototype specification (C), and an exemplary template file (F).

Extending *ProET*

Apart from just adapting existing system templates/designs, it is also possible to extend the tool by defining entirely new elements. Thereby it has to be differentiated between rather simple extensions—such as new, XML-based prototype specifications—and more sophisticated extensions regarding entirely novel components—for example, progress indicators or the like as separate elements.

Regarding the definition of a new prototype, simply a new XML specification is created, using the available set of tags and corresponding attributes. New/adapted String Templates for components can be introduced by providing a corresponding `type` attribute within the dialog tag and a corresponding String Template file; the `type`−value then is matched by a certain mechanism with potential templates until the most appropriate one is found; due to space reasons, the entire naming-and-matching mechanism is not explained in detail here. Finally, to modify not the basic form of prototype but only its UI design, additional CSS files may be created and just referenced in the `<dialog...>` tag of the specification. This permits an easy exchange and comparison of several design options.

The extension of the tool by entirely new components, on the other hand, also is possible, yet more complex. This additionally requires an adaption of the underlying Java code (e.g., create new container classes for the element, adapt the XML parsers to correctly parse that new elements, and so on). The details of this entire extension process, however, would fall out of the scope of this paper.

4 Experiences with ProET

For a preliminary assessment of the capabilities of *ProET*, we recreated knowledge systems that have been developed by our department in the past. Thereby, the first goal was to match those original systems as closely as possible. Figures 3 and 4 present the outcome of replicating two rather different systems; each figure shows both the original system in the background (A) and the prototype in the foreground (B). In the following, we first shortly introduce each system and summarize the insights regarding its replication with *ProET*. Based on that, we discuss the scope and limitations of the tool in the subsequent section.

4.1 Knowledge System Replication

The *Consultation on Rheumatic Disease*, Figure 3, served as the first case study. Based on the entered symptoms, that system consults the user as to whether a rheumatic disease is probable. Basically, a questionary-style is implemented, meaning that the system presents more than one question at a time to the user. Thereby, a certain coloring metaphor is applied for supporting an answering of the questions in the most reasonable sequence: Questions, that are already answered are colored gray; not yet answered questions are colored yellow, and the suggested next question is colored green.

This coloring-based interactivity is also mirrored in the prototype. There, JavaScript-based techniques are used to change the coloring according to the user's actions. Thus, the basic system interaction and styling can be realized by the means of the prototyping tool quite well. In some minor aspects, however, the prototype differs from the original system. First, a seemingly more loosely assembled interface appearance; whereas in the original system kind of a (visible) table-based layout was used, the prototype builds on a CSS-based layout, defining questions as separate elements and rendering them without using tables at all. If desired, however, it is with some effort possible, to adapt the template files as to also use (visible) tables for element arrangement. Furthermore, the original system provides a progress bar at the top of the dialog, indicating feedback on the proportion of already processed questions. Such feedback components are currently not yet included in *ProET*—by using JavaScript techniques, and by defining a corresponding component type and respective templates/styles, such or similar interactive feedback elements could be added in the future.

As a second case study, we chose the *Labour Legislation Consultation*, that in contrast to the rheumatic consultation implements a fundamentally different, hierarchy-based interaction style. There, the problem to solve—in this case the question, whether an employment contract was terminated legitimately—is displayed as the top element in the hierarchy (see Figure 4) and its current rating (e.g., established/suggested/excluded) is indicated by its coloring. On the next hierarchical sub-layer, questions are displayed that help to clarify that problem—in the example, the second item from the top "Compliance with form...", and all items on the same hierarchical level. If reasonable, those questions are further subdivided—for example, question "Dismissal was not prohibited..." (third item from the top) is subdivided into eight further questions, that describe more detailed aspects of the parent question. Thus, the user can choose, whether to answer the more abstract questions, or rather the more refined ones. Based on the provided answers, the system calculates ratings regarding the problem statement; those ratings of the sub-questions are accumulated into one rating, that is then presented for the parent question—this propagation proceeds up through the complete hierarchy to the main problem statement/solution.

Figure 4 shows, that this system type is matched well by the tool. Also the necessary interactivity—unfolding sub-hierarchies of questions by mouse-click, coloring the questions depending on the answer, and accumulating/propagating solution states throughout the hierarchy—is supported. One minor difference between the original system and the prototype again concerns the interface design: First, the "+" and "-" signs—indicating whether a question can be further subdivided—are not integrated in *ProET*. Also, the coloring of the questions does not end with the last character of the question as in the original, but spans a defined width.

4.2 ProET: Scope and Limitations

First replication case studies so far revealed the need to extend *ProET* with further components and templates, as to be able to recreate the chosen initial set of knowledge systems completely. Examples are aforementioned feedback components (e.g., progress bars), or further options/templates regarding the general layout (e.g., table-based, or multi-column layouts). Despite such minor shortcomings, *ProET* currently comprises a reasonable set of widgets, as well as exemplary prototype specification files, that enable the near-complete recreation of many knowledge systems, developed by our department in the past. Their replication, as well as adapting the given system specifications and corresponding template/style files for examining alternative designs, was rather unproblematic.

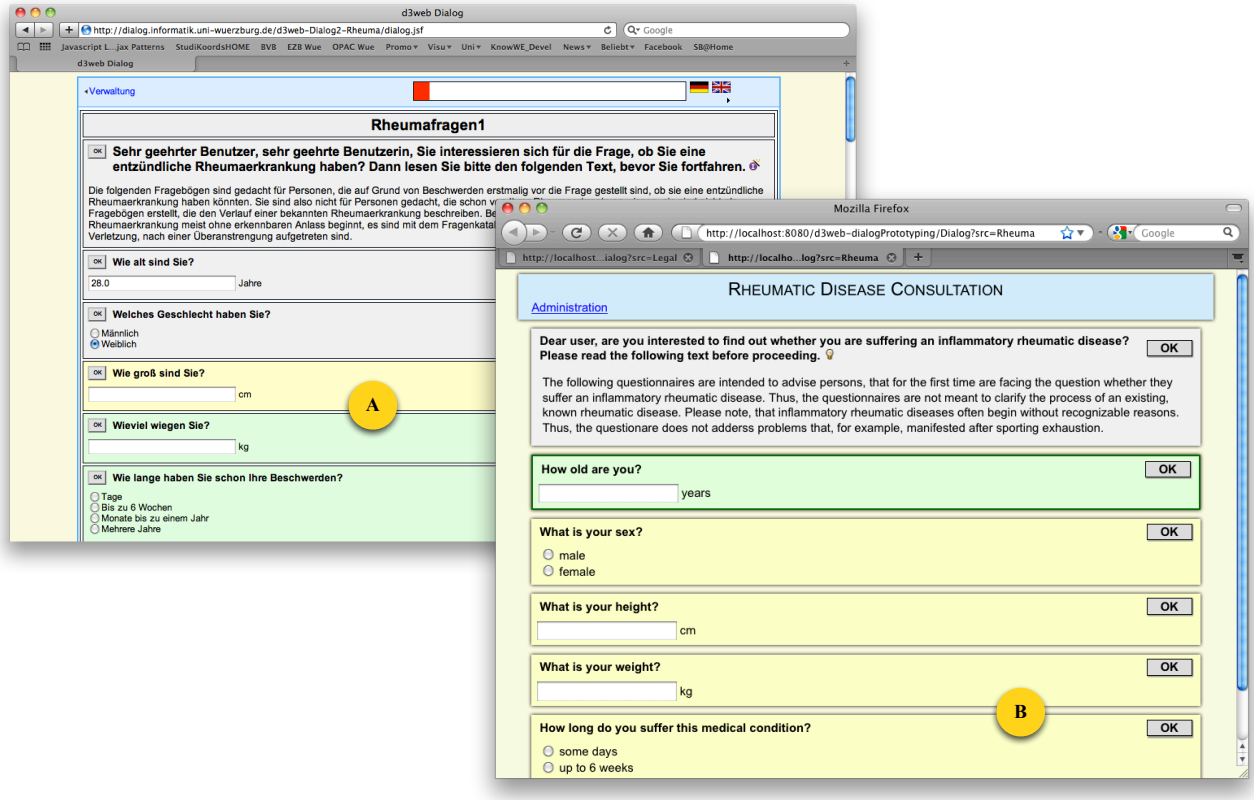


Figure 3: Consultation on rheumatic diseases—an example of a standard questionary-style consultation system. Original system (A, in german) and recreated prototype (B)

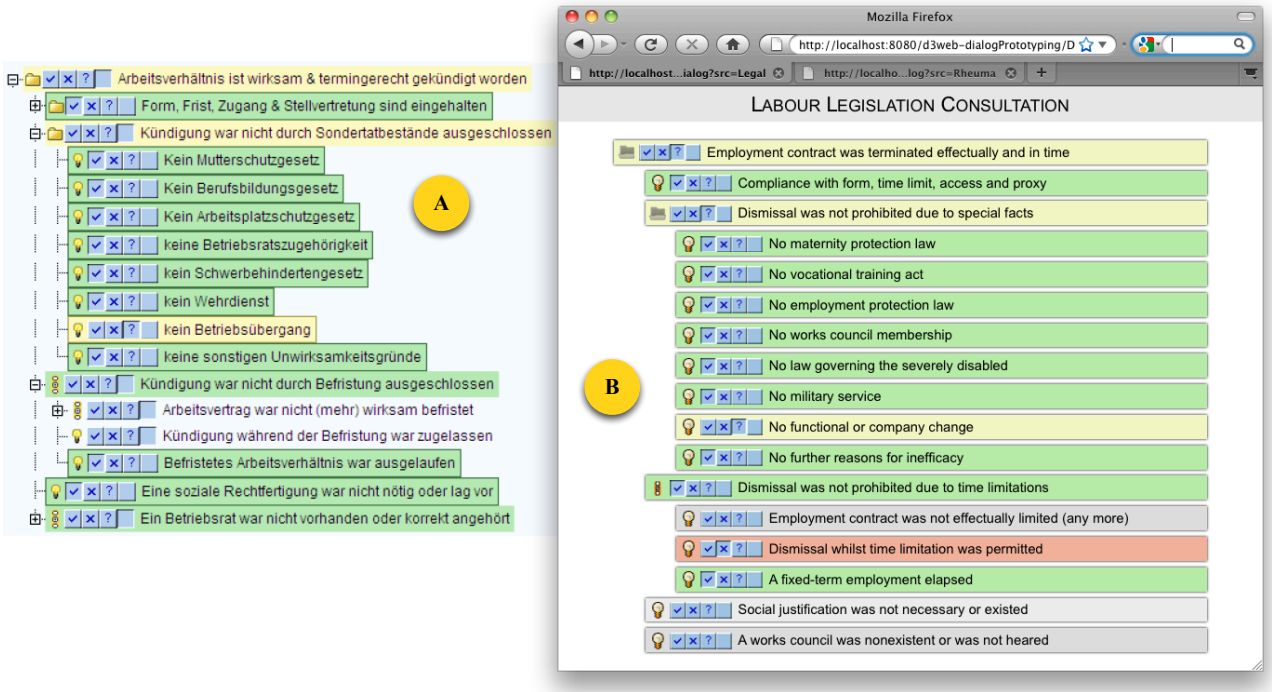


Figure 4: Labour legislation consultation—an example of a hierarchical-style consultation system. Original system (A, in german) and recreated prototype (B).

Yet, the different KS types and by default available widgets that can be prototyped without (major) tool extensions is limited. This is due to the fact that we explicitly chose web-based consultation and documentation systems for defining the initial set of features supported by the tool. Apart from the static widgets, so far also only selected interaction forms are supported. This mainly enables the creation of two basic system styles at the moment: Questionary-based (Figure 3) and navigable hierarchy-based (Figure 4). Those elementary styles can be adapted with regards to various aspects, such as grouping questions into questionnaires, optionally showing side panels that for a direct navigation of the pages, the presentation of solutions and their derivation states, or different forms of designing header and footer elements.

We are aware, that there surely exist other equally relevant knowledge system types and respective designs developed outside our department; yet, a more comprehensive investigation of such external systems, the identification of additional, fundamental KS components, and the appropriate extension of *ProET* is subject of further research.

5 Conclusion

In this paper, we introduced the tool *ProET* for developing knowledge system prototypes. We introduced a tailored process model for pattern-based, prototyping- and usability-integrating KS Engineering, and we discussed potential benefits as well as how the approach can be applied for pragmatically promoting and conducting respective projects.

First experiences with *ProET* revealed the need of its further extension. Apart from systems created by our department, also externally developed knowledge systems will be examined as to identify further relevant components and interactions; a more extensive classification of fundamental elements—in terms of a widget "language"/library—will be defined, leading to an extension of *ProET* as to match that "language". Also, prototyping with ProET is currently purely text-based. Extending the tool to allow also for a more visual form of prototyping—e.g., assembling prototype elements via drag & drop—is another interesting research issue, as this provides the chance to render the prototyping process per se much more intuitive.

Another open question is, whether usability evaluation components can and should be directly integrated into *ProET* (e.g., integrating some tailored logging mechanism to track the "usage" of the prototype). A further idea is incorporating tailored usability guidelines/heuristics into the tool in the form of interactive questionnaires, that can be optionally rendered integrated with the prototype, enabling its rather straightforward evaluation.

Also, the direct linking of the *d3web* toolkit[2] to *ProET* is under way. *d3web* facilitates the development of deployable knowledge bases, thereby supporting various problem-solving methods (e.g., heuristic rules, or set-covering models). This coupling first enables the integration of deployable knowledge bases with *ProET*, permitting an easy investigation, which KS type and corresponding (interaction) design is suitable for a given knowledge base—e.g., developed in the course of actual projects—or also more generally, for a specific knowledge representation. The long-term objective is to extend UI prototypes into productive knowledge systems with no or minimum additional effort.

[2] http://d3web.sourceforge.net/

References

[1] J. Angele, D. Fensel, D. Landes, R. Studer, Developing Knowledge-Based Systems with MIKE, Automated Software Engg. 5 (4) (1998) 389–418.

[2] J. Baumeister, D. Seipel, F. Puppe, Agile development of rule systems, in: Giurca, Gasevic, Taveter (eds.), Handbook of Research on Emerging Rule-Based Languages and Technologies: Open Solutions and Approaches, IGI Publishing, 2009.

[3] Bäumer, Dirk and Bischofberger, Walter R. and Lichter, Horst and Züllighoven, Heinz, User Interface Prototyping—Concepts, Tools, and Experience, in: ICSE '96: Proceedings of the 18th international conference on Software engineering, 1996, pp. 532–541.

[4] M. Beaudouin-Lafon, W. Mackay, Prototyping tools and techniques, in: The human-computer interaction handbook: fundamentals, evolving technologies and emerging applications, L. Erlbaum Associates Inc., Hillsdale, NJ, USA, 2003, pp. 1006–1031.

[5] C. Floyd, A Systematic Look at Prototyping, in: Approaches to Prototyping, Springer-Verlag New York, Inc., 1984.

[6] M. Freiberg, J. Baumeister, A survey on usability evaluation techniques and an analysis of their actual application, Tech. Rep. 450, Institute of Computer Science, University of Würzburg, Germany (2008).

[7] M. Freiberg, J. Baumeister, F. Puppe, Interaction pattern categories—pragmatic engineering of knowledge-based systems, in: Proceedings of the 6th Workshop on Knowledge Engineering and Software Engineering (KESE-2010) at the 33rd German Conference on Artificial Intelligence, 2010.

[8] H. Lichter, M. Schneider-Hufschmidt, H. Züllighoven, Prototyping in industrial software projects—bridging the gap between theory and practice, in: ICSE '93: Proceedings of the 15th international conference on Software Engineering, 1993, pp. 221–229.

[9] Y.-K. Lim, E. Stolterman, J. Tenenberg, The anatomy of prototypes: Prototypes as filters, prototypes as manifestations of design ideas, ACM Trans. Comput.-Hum. Interact. 15 (2) (2008) 1–27.

[10] J. Lin, J. A. Landay, Employing patterns and layers for early-stage design and prototyping of cross-device user interfaces, in: CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems, 2008, pp. 1313–1322.

[11] M. McCurdy, C. Connors, G. Pyrzak, B. Kanefsky, A. Vera, Breaking the fidelity barrier: an examination of our current characterization of prototypes and an example of a mixed-fidelity success, in: CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems, 2006, pp. 1233–1242.

[12] M. R. McGee-Lennon, A. Ramsay, D. McGookin, P. Gray, User evaluation of OIDE: a rapid prototyping platform for multimodal interaction, in: EICS '09: Proceedings of the 1st ACM SIGCHI symposium on Engineering interactive computing systems, ACM, New York, NY, USA, 2009, pp. 237–242.

[13] T. Memmel, H. Reiterer, A. Holzinger, Agile methods and visual specification in software development: a chance to ensure universal access, in: UAHCI'07: Proceedings of the 4th international conference on Universal access in human computer interaction, Springer-Verlag, Berlin, Heidelberg, 2007, pp. 453–462.

[14] J. Nielsen, Iterative User Interface Design, IEEE Computer 26 (11) (1993) 32–41.

Modeling of Diagnostic Guideline Knowledge in Semantic Wikis

Reinhard Hatko, Jochen Reutelshoefer, Joachim Baumeister, and Frank Puppe
Institute of Computer Science, University of Würzburg, Germany
{lastname}@informatik.uni-wuerzburg.de

Abstract

Knowledge acquisition for diagnostic knowledge systems is a complex and tedious task. In particular, the formalization of diagnostic guideline knowledge is challenging for the contributing domain specialists. In this paper, we introduce the formal representation language DiaFlux, that is simple and easy to use on the one hand. On the other hand it allows for the definition of executable clinical protocols, that can solve valuable tasks being executed in the clinical context. Further, we describe a wiki-driven development process using the stepwise formalization and allowing for almost self-acquisition by the domain specialists. The applicability of the approach is demonstrated by a project developing a protocol for sepsis diagnosis and treatment by a collaboration of clinicians.

This paper is currently under submission at the Open Knowledge Models Workshop at EKAW 2010.

1 Introduction

In recent years, knowledge engineering research has been heavily influenced by the emergence of Web 2.0 applications, such as wikis, blogs, and tagging systems. They provide a simplified access and a light-weight approach for knowledge acquisition. Furthermore, those systems usually allow for a distributed and (often) collaborative development process. One of the most popular examples is the wide-spread use of *wikis* as flexible knowledge management tools, both in personal life and business environments. In contrast to standard web applications the content of a wiki page can be created and modified by clicking an (often mandatory) *edit* button located on the web page. Due to the simple markup (less verbose than HTML), users are capable to author the content easily. Wikipedia is certainly the most popular example, where informal *world knowledge* is created and updated by a wiki. Introducing the *semantic interpretation* of wikis, the development of Semantic Wikis [1] allows for a more formalized definition of the knowledge. Today, Semantic Wikis are mainly used for collaborative ontology development, by providing a flexible, web-based interface to build semantic applications.

The main benefit of Semantic Wikis is their possibility to interweave different formalization types of knowledge in the same context. That way, ontological concept definitions are mixed with free text and images within the wiki articles. Such tacit knowledge often serves as documentation of the development process or as pursuing additional information not representable in a more formal manner. In detail, we call the interweaving of implicit and formal elements of knowledge and their interaction in the knowledge engineering process the *knowledge formalization continuum* [2]. The knowledge formalization continuum emphasizes that usable knowledge ranges from very informal representations—such as text and images—to very explicit representations—such as logic formulas or consistency-based models. The metaphor frees the domain specialists and knowledge engineers to commit to a particular knowledge formalization at an early stage of the development project, but offers a versatile understanding of the formalization process. Present Semantic Wiki implementations serve as ontology engineering tools with some support of rules, thus covering a wide range of the knowledge formalization continuum.

In this paper, we introduce the Semantic Wiki KnowWE, that was designed to build decision-support systems, and we propose the graphical language DiaFlux, for modeling of clinical protocols: The contributions of this language are its simple application for developing decision-support systems, since it only provides a limited number of intuitive language elements. Due to its simplicity it is possible to be used by domain specialists and thus ease the application in the knowledge engineering process. Albeit its simplicity, a rich set of diagnostic elements can be integrated into the language, that are required to build sophisticated (medical) knowledge bases. Furthermore, the language allows for the incorporation of less explicit knowledge elements when needed, and thus follows the ideas of the knowledge formalization continuum. To allow for comfortable development of DiaFlux models, we introduce a visual editor integrated into the Semantic Wiki KnowWE.

The rest of the paper is organized as follows: Section 2 briefly introduces the Semantic Wiki KnowWE and discusses the integration of strong problem-solving knowledge into the context of a Semantic Wiki. Also, some knowledge engineering aspects, such as the organization of a distributed knowledge base over a wiki, are discussed. The procedural knowledge representation language DiaFlux is introduced in Section 3. Also the integration of the language into the Semantic Wiki and development process including stepwise formalization is described. Currently, the approach is evaluated by the development of a medical decision-support system. We describe the essentials of this case study in Section 4. The paper is summarized and concluded in Section 5, also giving an outlook for future work.

2 KnowWE in a Nutshell

Wikis became famous as a successful means for knowledge aggregation in an evolutionary 'self-organizing' way by possibly large and open communities without detailed project management about contributions. Semantic Wikis extend the wiki approach by adding formal representations of the wiki content. This extension allows for two different perspectives of the use of Semantic Wikis [3]:

- **Knowledge formalization for wiki:** Here, the informal content of the wiki is in the foreground. The formalization used to help organization, navigation, and presentation of the content. Formal allowing for semantic navigation and search or the generation of new aggregated views on content, e.g., by inline queries.
- **Wiki for knowledge formalization:** In this case, the formalized knowledge base is the goal of the application. The knowledge base is created by the use of the well-known wiki authoring metaphor. One example is the use of a Semantic Wiki as a collaborative ontology engineering tool. The informal content here is used as description and documentation of the formal concepts and relations.

The wiki-based knowledge engineering approach described as well as the system introduced in this paper clearly focus on the latter perspective: The wiki is used as a tool for creating and maintaining formal concepts, formal relations, and informal knowledge containing description as documentation for these. While many Semantic Wikis provide means to create and populate light-weight ontologies, the approach can be generalized to create any kind of formal knowledge bases, e.g., for decision-support systems. The Semantic Wiki KnowWE provides methods to capture and execute problem-solving knowledge. Therefore, a problem-solving layer has been included on top of the ontological layer, defining findings that describe the currently running problem-solving session. KnowWE is designed showing only minimal modifications with respect to look and feel allowing for "backward compatible" use like a normal wiki. Figure 1 shows an article about *Bad Ignition Timing* as part of a car-diagnosis example wiki. It contains textual descriptions of the concept forming a solution of the diagnostic wiki knowledge base. Embedded in this informal content, it contains rules (1) defined by specific rule markup forming the formal knowledge deriving the concept *Clogged Air Filter* as solution of the problem-solving session. Further, KnowWE provides means for testing the knowledge, e.g., by answering popups (2) that are defined in the page content. The status of the derived solutions of the current problem-solving session is shown in the left panel (3). At the top of the formal knowledge block different icons provide additional features like starting a problem-solving session as a guided interview in an external dialog frame or download of the generated knowledge base (4). The problem-solving knowledge (rules in this example) is entered in textual format by specified markup using the standard wiki edit view, which is processed by the wiki engine and translated to an executable knowledge representation. Each time the content is edited, with the page save action the formal knowledge sections are processed by the wiki engine, updating the executable knowledge base accordingly. Different knowledge formalization patterns (e.g., heuristic rules, decision-trees, set-covering knowledge) are supported by the system and can be captured by the use of various markups [4].

Following the *freedom of structuring* principle proposed by wikis, the system does not put up any constraints where formal knowledge should be defined. Objects and formal relations can be used on any wiki page at any location using the markup defined by the current system settings. KnowWE also provides components to test the current version of a knowledge base. For automated testing of the knowledge base behavior, KnowWE also allows for the definition for test-cases, which can be executed after knowledge base modifications. For the creation of variants of a knowledge base the system provides a flexible include mechanism allowing to include arbitrary knowledge elements into a new page forming a new compilation of the overall knowledge corpus (e.g., for fault diagnosis in deviating production series in technical domains).

Wiki-based Knowledge Formalization The major strength of wikis is the general low barrier for contribution due to its simple editing mechanism. However, the definition of a formal knowledge base using textual markups is a complicated task. Autonomous contribution by domain specialists still can be achieved using a stepwise formalization process. Employing the metaphor of the knowledge formalization continuum at first informal knowledge describing the domain knowledge is inserted into the wiki. This can be done by domain specialists not demanding special knowledge engineering experiences. For this purpose also already existing documents can be imported into the wiki as startup knowledge that can be refined manually. After short training sessions discussing and modifying example knowledge bases the knowledge formalization task is started as an incremental process: The domain specialists, not completely familiar with the provided markups and the formalisms, at first formulate the knowledge that is right now only given as informal description, in pseudo-code style inspired by the markup. In cooperative sessions with the knowledge engineers this knowledge is discussed and transformed into correct syntactical shape. This proceeding allows for autonomous contributions, although if not yet completely formalized, by the domain specialists. The web-based collaborative access provided by wikis supports this evolutionary process.

3 DiaFlux - Modeling Clinical Care Processes in Semantic Wikis

This section first describes our application scenario, then a short insight about guideline models in the diagnostics domain is given. Following, we introduce our representation language for clinical protocols, called *DiaFlux* .

3.1 Application Scenario

Clinical guidelines have shown their benefits by providing standardized treatment based on evidence-based medicine. Many textual guidelines are readily available and also shared through the internet, but rely on the proper application by the clinician during the actual care process. While clinical guidelines are mostly textual documents, clinical protocols are an implementation of them, offering a more specific procedure for diagnosis and treatment in a given clinical context [5]. Much effort has been put into the development of formal models for computer-interpretable guidelines (CIGs). Clinical decision-support systems that execute CIGs support the clinician in his decision-making

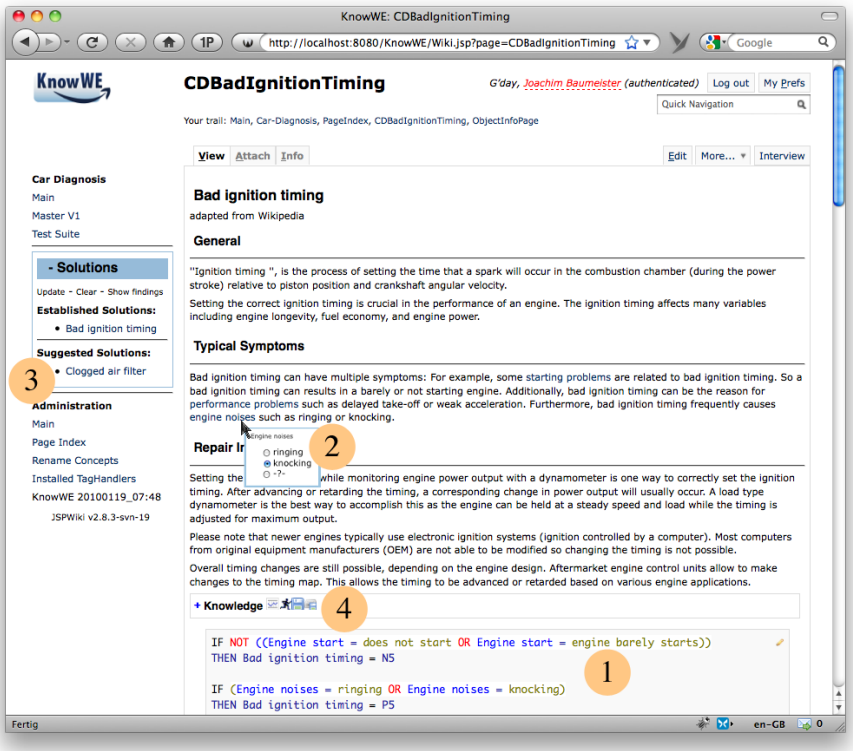


Figure 1: A wiki page of the car diagnosis example wiki containing formal and informal knowledge about the concept *Bad Ignition Timing*.

at the point of care. In the variety of CIG models, each has its own focus, e.g. GLIF [6] focuses on the shareability of guidelines between various institutions, while PRO*forma* [7] focuses on assisting patient care through active decision support [8].

The work presented in this paper is conducted within the project "CliWE - Clinical Wiki Environments"[1]. We investigate languages, tools and methodologies to collaboratively build CIGs by domain specialists themselves. The requirement concerning the language is the development of an explicit and executable representation of diagnostic knowledge for active decision-support systems. Furthermore, we create a development process for simple and effective knowledge acquisition by domain specialists. Finally, the completed knowledge bases will be exported into mixed-initiative systems, that cooperate with the clinical user during the care process.

3.2 Modeling Clinical Processes

For the representation of the tasks, that have to be carried out during the process of care, guideline languages employ different kinds of Task Network Models [9]. They describe decisions, actions and constraints about their ordering in a guideline plan. Often, flowcharts are the underlying formalism to explicitly express control flow, at least at an abstract level. GLIF for example, takes a multi-level approach, on three levels of abstraction, a *conceptual*, a *computable* and an *implementable level*. The conceptual level is modeled using flowcharts for interpretation by humans, but can not be executed by decision-support systems. The formal specification is provided at the next lower - the computable - level. On the implementable level a guideline can be tailored to a specific institution, e.g. by the definition of mappings to patient information systems.

For the specification of a clinical protocol, two kinds of knowledge have to be effectively combined, namely declarative and procedural knowledge. While the declarative part encompasses the facts and their relationships, the procedural one reflects the knowledge about how to perform a task, i.e. deciding which action to take next. In diagnostics the declarative knowledge particularly consists of the terminology, i.e., findings, solutions, and sometimes also treatments and their interrelation. The procedural knowledge for diagnostics in a given domain is responsible for the decision which action to perform next (e.g. in patient care: asking a question or carrying out a test). Each of these actions has a cost (e.g. monetary or associated risk) and a benefit (for establishing or excluding currently considered solutions) associated with it. Therefore, the choice of an appropriate sequence of actions is mandatory for efficient diagnosis and treatment.

We therefore propose a knowledge representation for clinical protocols called *DiaFlux* . It is designed to be a sufficiently expressive knowledge representation for executable clinical protocols, yet intuitive enough for the self-acquisition by domain specialists.

3.3 Modeling with DiaFlux

This paper introduces the formalization of clinical protocols with *DiaFlux* . By combining well-known elements from flowcharts (i.e. nodes and edges) with the tasks that are carried out during diagnostic problem-solving, it allows for a uniform and intuitive acquisition of declarative and procedural knowledge. As known from flowcharts, the nodes represent actions, that have to be executed, and the edges connecting those nodes the order of the execution.

[1]Funded by Drägerwerk AG & Co. KGaA, Lübeck, Germany, 2009-2011.

An edge can be guarded by a condition, that controls the transition to the subsequent node. The nodes represent actions like performing a test or evaluating a diagnosis. The guards are attached to edges and they control the process by checking the state of the declarative knowledge, e.g. the value of a finding or the evaluation of a solution.

Clinical protocols, that are formalized with DiaFlux, are intuitively understandable and more easily maintainable, as the sequence of actions is made explicit. Furthermore, due to the semantics of an underlying application ontology it directly is executable.

Design Goals

When designing a formalism – especially when aimed at being used by non-computer-scientists – a trade-off has to be made between its *expressiveness* and its *usability/understandability*. In our approach we favor usability over expressiveness, offering a minimal set of primitives, though expressive enough for the targeted scenario. Besides, the following goals were pursued during the design of DiaFlux:

1. *Modularity*: To alleviate the reuse of (parts of) formalized knowledge, DiaFlux models are intended to be reused in different contexts. The modularization also helps to improve the maintainability of the knowledge base.
2. *Repetitive execution of subtasks*: Online monitoring involves the continuous observation of sensory data to detect fault states and initiate corrective action. Therefore particular actions need to be performed in an iterative manner.
3. *Parallelism*: Subtasks with no fixed order and dependency can be allocated to additionally spawned threads of control, and thus allow for their parallel execution. Expressing parallelism is especially necessary for mixed-initiative diagnosis, in which human and machine initiated examinations are carried out concurrently.
4. *Testability*: The evaluation of a knowledge base is an essential step prior to its productive use. We provide basic functionality for empirical testing and anomaly checks tailored to DiaFlux models.

Language Description

Wang et al. [10] studied several guideline representation models and identified common primitives for guideline creation. These were categorised as *actions*, *decisions* and for the representation of *patient state*. Actions denote specific clinical tasks like collecting data or clinical intervention. Decisions represent clinical decision making and guide the course of the care process. Patient states represent a patient's clinical status in the context of guideline application.

As DiaFlux models are based on flowcharts, we identified a minimal subset of node types to incorporate the necessary primitives to express a clinical protocol. To express the procedural aspect of the protocol, nodes can be connected by edges. There are no different types of edges, but they can be labeled with different types of conditions. The actual type of the condition depends on the type of node the edge starts at. They are used to evaluate the declarative knowledge with respect to the observed findings, e.g. the outcome of a given test or the status of a diagnosis. To obtain the semantics necessary for executability, we rely on an application ontology as an extension to the task ontology of diagnostic problem solving [11]. The application ontology defines the declarative knowledge consisting of findings and their ranges, solutions and treatments.

In the following, we enumerate and informally describe the different node types, before we give a toy example of a DiaFlux model in the introduced car fault diagnosis domain:

- **Start node:** A start node does not imply an action itself, but is a pseudo-node pointing to the node that represents the first action to take. Multiple start nodes can be modeled to provide distinct entry points into one DiaFlux protocol.
- **Test node:** Test nodes represent an action for carrying out a single test on activation of the node at runtime. This may trigger a question the user has to answer or data to be automatically obtained by sensors or from a database. Furthermore, the acquired information refines the knowledge about the patient state.
- **Solution node:** Solution nodes are used to set the evaluation of a solution, based on the observed findings.
- **Wait node:** Upon reaching a wait node, the execution of the protocol is suspended until the given time has passed.
- **Composed node:** DiaFlux models can be hierarchically structured as already defined ones can be reused as modules, represented by a composed node. This fulfills the aforementioned goal of modularity.
- **Exit node:** An exit node terminates the execution of a DiaFlux model and returns the control flow to the superordinate model. To express different results of a model, several distinct labeled exit nodes are supported.
- **Comment node:** For documentation of a protocol, comment nodes can be inserted at arbitrary positions. Though, they can be connected by edges and so be used to create semi-formal guidelines. They do not represent an action and are ignored during execution.

Figure 2 shows a DiaFlux module for handling the problem area "Battery", which has been established by another - superordinate - module (not shown in Figure 2). It is embedded into the wiki article containing further informal information about batteries.

The execution of the module starts at the *start node* "Start" (1), which is pointing to the node "Battery voltage" (2). It is a *test node*, which asks the user for the actual voltage of the battery. As "Battery voltage" is a question with a numerical range, the conditions that can be modeled on the outgoing edges, are checks against disjoint intervals. In case the battery's voltage is high enough to start the car (> 12.5V), the fault may be a connection problem to the electrical system. So, the user is asked whether the terminals are clean. In case they are, the problem has to be the automobile self starter or its cabling. Then, the execution reaches a *solution node* that evaluates the solution "Damaged Starter" as suspected (3). An attached comment node gives a hint for further elaboration, pointing out, that also the cabling may be the fault case.

Having rusty terminals, a connection problem is likely. The instructions of how to clean the terminals are a self-contained module, that can readily be reused in this protocol. Upon reaching the *composed node* "Clean Terminals" (4) the execution of the according protocol is started and "Battery Check" is stalled until the subordinate one is finished. In this example, the module "Clean terminals" exclusively consists of instructions to the user, so

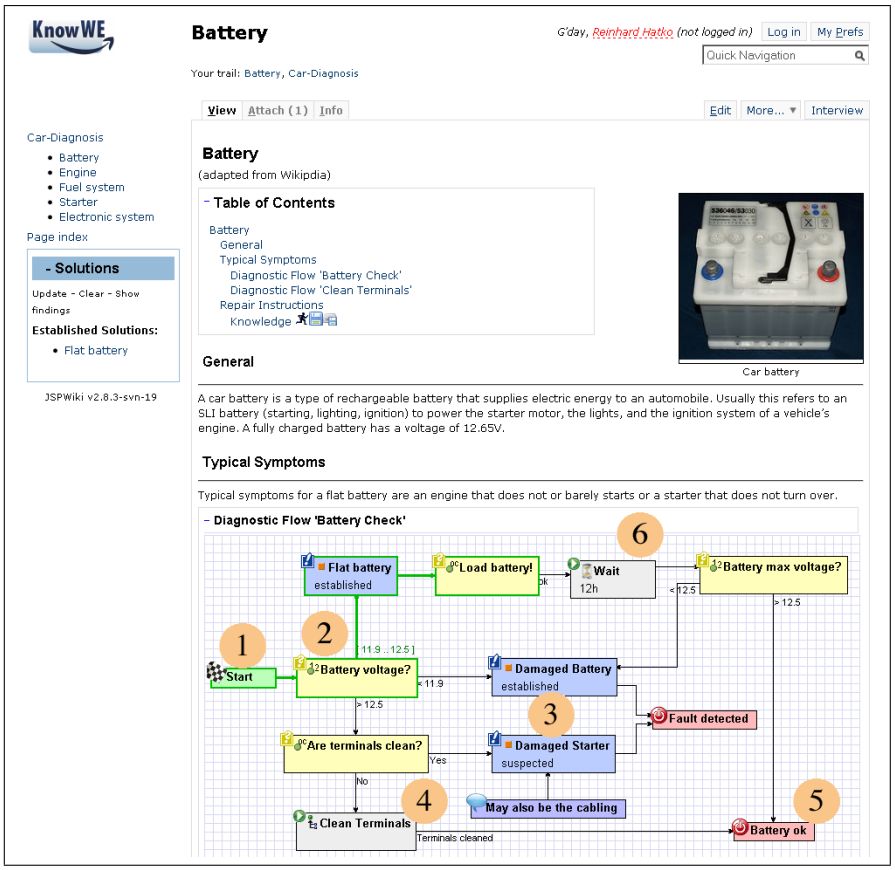


Figure 2: A guideline for diagnosing a car battery, embedded into a wiki article containing further information. The pathway of the current testing session is highlighted in green.

the only possible outcome of this procedure is "Terminals cleaned"(represented by an according exit node), which is the condition on the outgoing edge. After following the instructions and finishing the module, the *exit node* "Battery ok" (5) is activated, returning execution to the superordinate protocol (not shown in Figure 2).

In case the battery's voltage is too low to reliably start the engine (below 12.5V), different solutions can be established. If the voltage lies below 11.9V, then the battery is exhaustively discharged and considered broken. Therefore, the solution "Battery damaged" is established and the execution is ended by reaching the exit node "Fault detected". If the voltage is in the range between [11.9V, 12.5V], then it is probably too low for activating the self starter, but can be recharged. After establishing the solution "Flat Battery", the user is instructed to load the battery by the node "Load battery" (this basically is modeled by a one-choice question with the only one answer "ok"). Then the *wait node* (6) is activated suspending the execution for 12 hours. After this period has passed, the execution of the protocol is resumed. The user is asked for the actual battery voltage again, which is the battery's maximum voltage. If the voltage still is too low to start the car (< 12.5V), a "Damaged Battery" can again be established. If the voltage is sufficiently high, the taken exit node "Battery ok" indicates, that there must be another cause for the fault. After returning to the superordinate protocol, further steps can be taken to find the cause of the fault, e.g., a damaged engine. The outgoing edges in the superordinate protocol starting at the composed node representing "Battery Check" can decide how to proceed further, depending on the taken exit node.

Integration in KnowWE

We created an implementation of DiaFlux for the knowledge-based system d3web [12]. DiaFlux offers the possibility to model and execute protocols that employ declarative and inferential expressiveness provided by d3web.

An AJAX-based editor for DiaFlux is integrated into the Semantic Wiki KnowWE (cf. Figure 3), using its plugin mechanism [13]. The DiaFlux editor is on the one hand able to reuse ontological concepts that are readily available in the wiki's knowledge base. Those can simply be dragged into the flowchart. Depending on the type of object (finding, solution, DiaFlux model), a node of adequate type is created. On the other hand, the application ontology can be extended by creating new concepts from within the editor with an easy to use wizard. The model's source code is encoded in XML and integrated into the corresponding wiki article and saved and versioned together with it. This allows for further documentation of the protocol by tacit knowledge in the article. When the article is displayed in a web browser, the model visualization is rendered, instead of displaying its XML source code.

A related wiki environment for the collaborative creation of clinical guidelines is the Modelling Wiki (MoKi) [14], based on Semantic Media Wiki [15]. Originally it was designed for the creation of enterprise models using a visual editor, but it also has been used in the Oncocure project [16] to acquire clinical protocols for breast cancer treatment. Therefore, templates were defined within the wiki and later their content was exported into skeletal Asbru plans [17]. Though MoKi's visual editing capabilities for business pro-

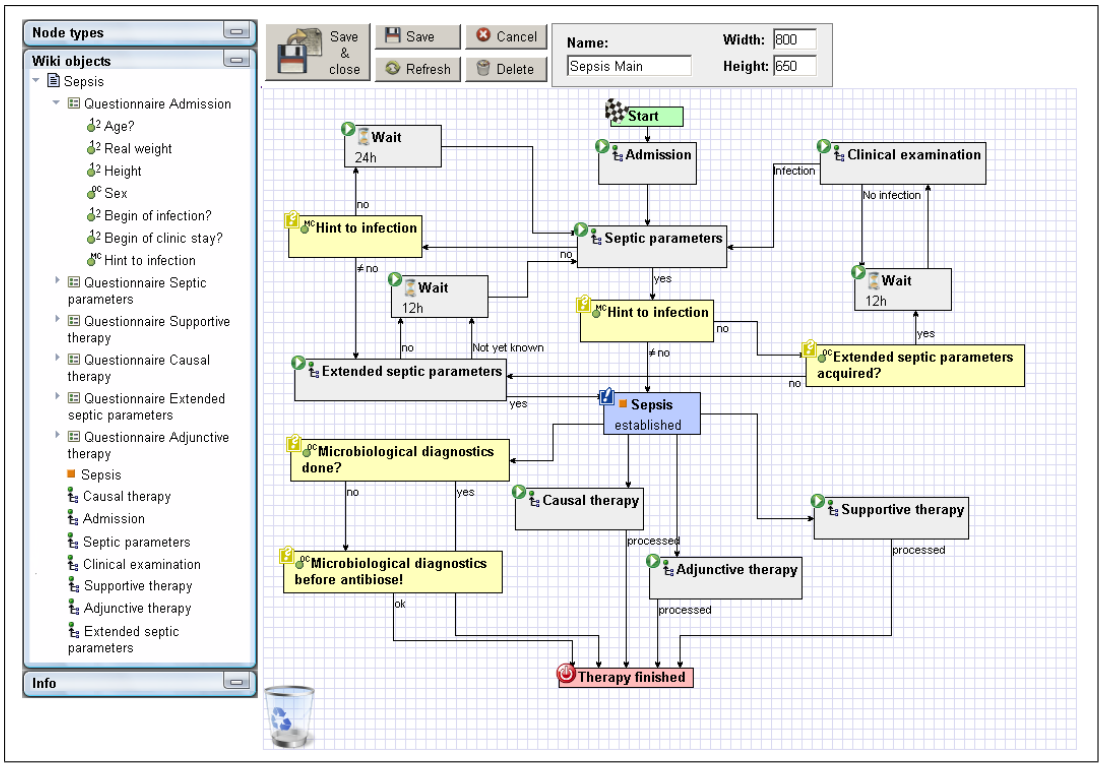


Figure 3: The main module of the sepsis diagnosis and treatment protocol, opened in the AJAX-based editor.

cesses, they were not employed to graphically model guidelines. Furthermore, the created Asbru plans are currently not executable within the wiki.

Development Process

For the development of DiaFlux models we propose the idea of the knowledge formalization continuum [2]: At first, informal information can be collected in wiki articles, e.g. about goals of a protocol. During the next step, a first semi-formal flowchart can be created using only comment, start and exit nodes and connecting edges. At this stage of formalization, the flowcharts can not be automatically executed, but "manually". For testing purposes the user can run through the flowchart by clicking on that outgoing edge of the active node, he wants to continue the pathway on. The taken pathway is highlighted for easier tracking. This especially is useful, when parallelism or hierarchically structured protocols are involved. The last step is the formalization into a DiaFlux model and the creation of the application ontology, resulting in a fully formalized and executable knowledge base. By following this process of gradual refinement, the entry barrier for domain specialists is quite low, while knowledge acquisition can start from the beginning.

Graphical modeling languages as a mediator to extract knowledge are also used in the *IT-Socket* of the research project *plugIT* [18]. Compared to our process of gradual refinement within the same modeling language, the IT-Socket employs semi-formal graphical models created by domains specialists, that are later formalized on a different level of abstraction.

Development Tools

Collaborative development requires to track the changes of all participants. Therefore, a frequent task is to compare different versions of a wiki article. For this purpose in general, wikis provide a textual diff comparing two versions of an article. As a diff of the XML source code is not very helpful for comparing a visual artifact like a flowchart, a more understandable diff is provided. On the one hand, a textual summary of the added, removed, and changed nodes and edges is generated. On the other hand, the previous and the current version of the DiaFlux model are shown next to each other, highlighting the changes in different colors for easy comparison, e.g. removed items are red in the previous version, added items are green in the current one, and changed items are highlighted in both versions.

After creating a knowledge base in KnowWE, a test session can directly be started from the wiki article containing it. Having used DiaFlux models, the current state of the protocol throughout the session can be observed. The traversed pathway through the flowchart is highlighted, in a similar manner as in the visual diff (cf. Figure 2). This immediate feedback considerably eases the interactive testing of the knowledge base.

4 Case Study

In the context of the project "CliWE" we used a prototype of the clinical wiki environment for the development of a protocol covering the diagnosis and therapy of sepsis. Sepsis is a syndrome of a systemic inflammation of the whole body. Despite the high mortality of this critical illness (30 to 60%) there are two main problems in sepsis therapy. First, it is essential to recognize that a patient fulfills sepsis criteria and second, if sepsis is diagnosed a complex medical therapy has to be initiated quickly. Today, so called patient data management systems are available in many intensive care units. With these systems, medical data are electronically available. In this context a clinical decision support system may be a reasonable solution for the above outlined practical problems, monitoring all patients for sepsis and support the physician in the initiation of sepsis treat-

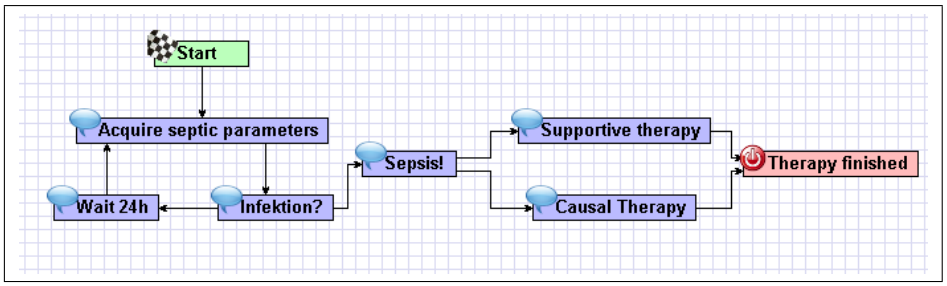


Figure 4: An early semi-formal version of the sepsis protocol.

ment.

The knowledge base was developed in accordance to the official guideline by the German Sepsis Society [19]. It is a textual guideline of about 80 pages describing the prevention, diagnosis, and therapy of sepsis. Our formalization of the guideline contains so far the diagnostics and parts of the therapy together with some common tasks for patient admission (cf. Figure 3). At the moment it contains about 50 nodes in eight modules with several possible pathways, depending on how the diagnosis can exactly be established and the course of the therapy. The upper part of the main DiaFlux model contains knowledge about the decision making and the lower part contains knowledge about the treatment.

The diagnosis task involves the assessment of up to eight clinical parameters (conducted in the modules "Septic parameters" and "Extended septic parameters") and an established or suspected infection. The monitoring is repeated until a sepsis can be established within different cycles depending on which parameters are acquired and their evaluation. If there is enough evidence to support a suspected sepsis, then a warning to the clinician is generated. If the clinician agrees with the conclusion, the diagnosis "Sepsis" is established and instructions for starting the therapy are given. The treatment for sepsis consists of the three bundles *causal therapy* (treating the cause of the infection), *supportive therapy* (stabilizing the patients circulation) and *adjunctive therapy* (supporting fighting off the infection). Those bundles are modeled as self-contained modules and reused as composed nodes in the main module.

Experiences

The knowledge acquisition mainly took place in two workshops, approximately six hours each, involving two domain experts. The DiaFlux editor was handled by a knowledge engineer, entering the knowledge artifacts provided by the domain specialists. The remaining participants followed the authoring process on a projector.

During the first session we followed the idea of the knowledge formalization continuum and started with textual descriptions of most modules. As a second step, we created semi-formal flowcharts giving an outline of the protocol, as exemplified in Figure 4. Next, we started to further formalize these flowcharts into executable DiaFlux models and to create the according declarative knowledge. The second session began with the acquisition of test cases of typical sepsis patients. As they were only informally entered in a wiki article and not executable so far, we stepped manually through the model by highlighting the correct pathway. The found inconsistencies were corrected during the second half of the session, together with further elaboration of the knowledge base. In a third session of about one hour, one of the experts created a small module by himself, while being observed by a knowledge engineer. The expert shared his screen using an internet screen sharing software and was supported in formalizing the knowledge and the usage of the DiaFlux editor.

Overall, the wiki-based approach showed its applicability and usefulness, as the combination of formal and informal knowledge and its gradual refinement was intensely used during the acquisition of the protocol and the test cases. Further, the developed knowledge base was accessible to all participants immediately after the workshops, as it took place in a password protected wiki, which can be accessed over the internet.

So far, the knowledge acquisition was conducted in workshops involving domain experts and knowledge engineers. After the initial workshops and the successful teleknowledge acquisition session, we are confident to proceed with further workshops, that require minimal support by the knowledge engineers.

5 Conclusion

This paper presented work in the context of the project "CliWE - Clinical Wiki Environments" for collaborative development and evolution of clinical decision-support systems. We introduced a language that can incorporate declarative and procedural diagnostic knowledge for modeling executable clinical protocols. Its main focus is simplicity for the usage by domain specialists. DiaFlux is integrated into the Semantic Wiki KnowWE to support the collaborative development by a community of experts. The case study demonstrated the applicability and benefits of the approach during the development of a clinical protocol for sepsis diagnosis and treatment. Due to the wiki-based approach the knowledge can evolve easily. It is accessible without depending on specialized software, as long as an internet connection is available. Furthermore, domain specialists can almost instantly start contributing. Formalization of the knowledge can then happen at a later time, after familiarizing with the semantics.

As next steps we plan the integration of refactoring capabilities into the editor, for the easier evolution of DiaFlux models. We will also enhance the tool support for the gradual formalization. As there rarely is only one single opinion in medicine, we will support different "medical schools" represented by the contributing experts by the possibility to engineer *variants* of DiaFlux models. In the future, we are planning the collaborative development by a community of experts, connected by KnowWE.

References

[1] Schaffert, S., Bry, F., Baumeister, J., Kiesel, M.: Semantic wikis. IEEE Software **25**(4) (2008) 8–11

[2] Baumeister, J., Reutelshoefer, J., Puppe, F.: Continuous knowledge engineering with semantic wikis. In: CMS'09: Proceedings of 7th Conference on Computer Methods and Systems (Knowledge Engineering and Intelligent Systems). (2009) 163–168

[3] Buffa, M., Gandon, F., Ereteo, G., Sander, P., Faron, C.: SweetWiki: A semantic wiki. Web Semantics **8**(1) (2008) 84–97

[4] Baumeister, J., Reutelshoefer, J., Puppe, F.: Markups for knowledge wikis. In: SAAKM'07: Proceedings of the Semantic Authoring, Annotation and Knowledge Markup Workshop, Whistler, Canada (2007) 7–14

[5] Hommersom, A., Groot, P., Lucas, P., Marcos, M., Martínez-Salvador, B.: A constraint-based approach to medical guidelines and protocols. In Teije, A.t., Miksch, S., Lucas, P., eds.: Computer-based Medical Guidelines and Protocols: A Primer and Current Trends. Volume 139 of Studies in Health Technology and Informatics. IOS Press (2008) 213–222

[6] Boxwala, A.A., Peleg, M., Tu, S., Ogunyemi, O., Zeng, Q.T., Wang, D., Patel, V.L., Greenes, R.A., Shortliffe, E.H.: GLIF3: a representation format for sharable computer-interpretable clinical practice guidelines. J. of Biomedical Informatics **37**(3) (2004) 147–161

[7] Fox, J., Johns, N., Rahmanzadeh, A.: Disseminating medical knowledge: the proforma approach. Artificial Intelligence in Medicine **14**(1-2) (1998) 157 – 182 Selected Papers from AIME '97.

[8] de Clercq, P., Kaiser, K., Hasman, A.: Computer-interpretable guideline formalisms. In ten Teije, A., Miksch, S., Lucas, P., eds.: Computer-based Medical Guidelines and Protocols: A Primer and Current Trends. IOS Press, Amsterdam, The Netherlands (2008) 22–43

[9] Peleg, M., Tu, S., Bury, J., Ciccarese, P., Fox, J., Greenes, R.A., Miksch, S., Quaglini, S., Seyfang, A., Shortliffe, E.H., Stefanelli, M., et al.: Comparing computer-interpretable guideline models: A case-study approach. JAMIA **10** (2003) 2003

[10] Wang, D., Peleg, M., Tu, S., Boxwala, A., Greenes, R., Patel, V., Shortliffe, E.: Representation primitives, process models and patient data in computer-interpretable clinical practice guidelines:: A literature review of guideline representation models. International Journal of Medical Informatics **68**(1-3) (2002) 59 – 70

[11] Baumeister, J., Reutelshoefer, J., Puppe, F.: KnowWE: A semantic wiki for knowledge engineering. Applied Intelligence (2010)

[12] Baumeister, J., et al.: The knowledge modeling environment d3web.KnowME. open-source at: http://d3web.sourceforge.net (2008)

[13] Reutelshoefer, J., Lemmerich, F., Haupt, F., Baumeister, J.: An extensible semantic wiki architecture. In: SemWiki'09: Fourth Workshop on Semantic Wikis – The Semantic Wiki Web (CEUR proceedings 464). (2009)

[14] Ghidini, C., Kump, B., Lindstaedt, S.N., Mahbub, N., Pammer, V., Rospocher, M., Serafini, L.: MoKi: The enterprise modelling wiki. In: ESWC'09: The Semantic Web: Research and Applications. Volume 5554 of LNCS., Springer (2009) 831–835

[15] Krötzsch, M., Vrandecić, D., Völkel, M.: Semantic MediaWiki. In: ISWC'06: Proceedings of the 5th International Semantic Web Conference, LNAI 4273, Berlin, Springer (2006) 935–942

[16] Eccher, C., Rospocher, M., Seyfang, A., Ferro, A., Miksch, S.: Modeling clinical protocols using Semantic MediaWiki: the case of the oncocure project. In: K4HelP: ECAI 2008 Workshop on the Knowledge Management for Healthcare Processes, University of Patras (2008) 20–24

[17] Miksch, S., Shahar, Y., Johnson, P.: Asbru: A task-specific, intention-based, and time-oriented language for representing skeletal plans. In: UK, Open University. (1997) 9–1

[18] Woitsch, R., Utz, W.: The IT-Socket: Model-based business and IT alignment. In Weghorn, H., Isaías, P.T., eds.: IADIS AC (1), IADIS Press (2009) 141–148

[19] German Sepsis-Society: Sepsis guideline. http://www.sepsis-gesellschaft.de/DSG/Englisch

Linked Data Games: Simulating Human Association with Linked Data

Jörn Hees, Thomas Roth-Berghofer, and **Andreas Dengel**
Knowledge Management Department,
German Research Center for Artificial Intelligence DFKI GmbH
and Knowledge-Based Systems Group, University of Kaiserslautern,
{firstname.lastname}@dfki.uni-kl.de

Abstract

Teaching machines to understand human communication is one of the central goals of artificial intelligence. Psychological research indicates that human associations are an essential requirement to understand human communication. In this paper the hypothesis is presented that simulating human associations with the help of Linked Data could improve text understanding capabilities of machines. To investigate whether human associations can be simulated with Linked Data, two preliminary problems are identified: (i) A reasonable ground truth for human associations is lacking and (ii) human associations have different strengths while Linked Data treats all triples equally and does not provide edge weights. To overcome these problems, two ideas for web games in accordance with Luis von Ahn's Games with a Purpose are proposed trying to turn the tedious acquisition processes into fun games. The resulting datasets are then to be used for quantitative comparisons of human associations and Linked Data.

1 Introduction

Since its introduction in 2001 the Semantic Web has gained much attention. In recent years, especially the Linked Open Data (LOD) project[1] contributed many large, interlinked and publicly accessible datasets, generating one of the world's largest, distributed knowledge bases. The accumulated amount of Linked Data can already be used to answer astonishingly complex questions (e.g., compiling a list of all musicians who were born in Berlin before 1900) or to provide additional information for selected concepts on a website (e.g., providing a short abstract and a thumbnail when hovering over the name Barack Obama).

In many ways Linked Data reminds of *spreading activation semantic networks* [Collins and Loftus, 1975]. Spreading activation semantic networks are successfully used in psychology to model human associations (e.g., thinking of *Barack Obama*, one will most likely also think of *USA*). Human associations are an important processing ability of our memory allowing us to retrieve related thoughts, traverse from one thought to another and thereby facilitate our way of thinking. They are also crucial for our understanding of everyday communication [Gerrig and Zimbardo, 2010, pp. 240ff] as they help us to build a context and resolve ambiguities. Hence it seems plausible that simulating such associations could help to improve text understanding capabilities of machines.

This work investigates the question if and how it is possible to simulate human associations with the help of Linked Data. A first analysis identifies two problems:

Currently, a quantitative comparison between Linked Data and human associations is impossible due to the lack of a reasonably large ground truth of human associations. Such a ground truth would consist of a large number of association pairs (e.g., (*Barack Obama*, *President of the US*)) collected from many different test persons. Collecting such a ground truth would allow us to answer questions such as: how large is the overlap between Linked Data and human associations? Nevertheless, due to the desired size, the acquisition would be infeasible with traditional approaches, such as paying test persons to record their associations.

The second problem is that human associations have different strengths, while Linked Data treats all triples equally and currently does not provide edge weights. Solving this problem would for example allow us to ask a machine to only show us the 20 strongest associations related to a resource, which in turn could be used to narrow down search spaces, use spreading activation algorithms in a meaningful way, or rank the results by association strengths. While several approaches try to rate triples by heuristics, none of them was compared to a dataset of human association strengths. Nevertheless, the acquisition of such a dataset would require us to assign weights to a very large number of Linked Data triples, which again would be infeasible with traditional approaches.

As both datasets need to be collected in order to investigate if it is possible to simulate human associations with Linked Data, two ideas for games in accordance with Luis von Ahn's Games with a Purpose [von Ahn and Dabbish, 2008] are proposed, turning the tedious process of entering associations or ratings into fun games.

The remainder of this paper is structured as follows: First the current state of the art is presented. Then a brief introduction into human communication is given explaining the important role played by associations in the language understanding process. In the following section human associations are conceptually compared to Linked Data, resulting in the two problems outlined above. For each of these problems a game idea is proposed, preceding the final conclusion and outlook.

2 State of the art

In recent years the research community devoted much attention to the Semantic Web [Berners-Lee *et al.*, 2001],

[1] http://esw.w3.org/SweoIG/TaskForces/CommunityProjects/LinkingOpenData

which proposed the vision of sharing information in the web not only to other humans, but also transforming it into a meaningful form for machines. Two of the most famous outcomes of these developments are the Resource Description Framework (RDF)[2] and the Web Ontology Language (OWL)[3]. Still, as manually providing information in these languages is quite cumbersome and seldom results in an immediate benefit for the authors, a few years went by without a reasonable amount of RDF being published on the web. Realizing this, a kind of grassroots development began to publish data which was already available in several centralized, corporate or governmental databases. Quickly guidelines were developed for publishing such data and interlinking it with others. With the ongoing help of volunteers and famous proponents, the so called Linked Open Data project was born, nowadays forming the so called Linked Data Cloud of interlinked datasets, including over 13.1 billion RDF triples as of November 2009. The content of this cloud, often called Linked Data, poses the world's largest distributed knowledge base and can be seen as a first challenge of the vision of the Semantic Web.

As more and more datasets were integrated into this Linked Data Cloud certain centralized points evolved, one of them being DBpedia[4]. The DBpedia team tries to automatically extract structured information from Wikipedia articles. In contrast to many other datasets, DBpedia represents very much knowledge across a large number of domains, which makes it very interesting for tying domains together. At the same time due to the automatic nature of the extraction, DBpedia also introduces a lot of errors into the Linked Data Cloud.

Even long before the Semantic Web started to evolve, psychological research has been very busy in the field of how human beings understand language. While the next sections go more into detail on spreading activation semantic networks [Collins and Loftus, 1975], it shall be mentioned here that they belong to the human semantic memory, which is a part of the so called explicit memory [Baddeley *et al.*, 2009, pp. 113–121].

As mentioned in the introduction, in this paper two games are proposed that try to turn the otherwise unfeasible work into fun games. This is motivated by Luis von Ahn's Games With A Purpose[5], which are part of a field called Human Computation. After the big success of the famous ESP Game [von Ahn and Dabbish, 2004], which turns the tedious process of labeling images into a fun game, and further games, a summary of design principles for Games With A Purpose [von Ahn and Dabbish, 2008] was published.

The proposed game ideas are especially related to the ESP Game, Verbosity [von Ahn *et al.*, 2006], Matchin [Hacker and von Ahn, 2009] and OntoGame [Siorpaes and Hepp, 2007]. Verbosity is a game which turns the widely known Tabu game into a game collecting common-sense facts. The game is of an asymmetric type, which means that there's a describer who gets a word that she has to describe to the guesser. The guesser can see the describer's output and guesses her input. In order to prevent cheating and to direct the collection of common-sense facts, the describer can not enter text freely, but has prepared snippets, which can be completed, such as "is a", "has a", "is the opposite of". If human associations were extracted from the facts collected with this game, they would be strongly biased towards these snippets. Besides this, the results of the game do not seem to be published. Matchin is a game, which presents pairs of images to the players and asks them which one their partner will prefer. The collected amount of relative votings is then used to globally rank the pictures. OntoGame can be seen as kind of the first game of this kind applied to Linked data. Nevertheless, it solves a very different task than the later on proposed games, namely finding out whether a Linked Data resource is an instance or a class and then trying to file them into a taxonomic structure.

[2] `http://www.w3.org/TR/rdf-primer/`
[3] `http://www.w3.org/TR/owl-ref/`
[4] `http://dbpedia.org`
[5] `http://www.gwap.com`

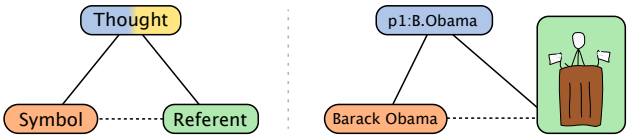


Figure 1: Semiotic Triangle after Ogden & Richards (left) and an example (right)

3 Human Communication

Human communication is the process of transporting information from one human being to another.[6] In such communication we can distinguish between *symbols*, THOUGHTS and `referents` as they can be visualized in the so called semiotic triangle (Figure 1).

This distinction allows us to see one of the main characteristics of human communication: Thoughts heavily depend on the respective person, and we are not able to exchange thoughts directly. A THOUGHT (e.g., P1:B.OBAMA) is someone's mental representation of some `referent` (e.g., `Barack Obama`, the one person with that name, currently being president). Instead of exchanging a thought directly, we are only able to exchange a *symbol* for the thought in written or spoken form (e.g., the two words *Barack Obama*). As a speaker or writer we then hope that the listener or reader of such a symbol finds an own thought that is sufficiently similar to our thought (see Figure 2, P2:B.O.), or creates a new thought for what we described.

Communication is not limited to the exchange of single thoughts, but is all about exchanging information (i.e., the connections between thoughts). As we can not exchange thoughts directly, we can only exchange information by a from of symbolic indirection. From a Semantic Web point of view, information can be expressed as simple statements. Each statement essentially is a simple sentence in the form of (subject, predicate, object)-triples, but instead of the usual symbolic form of a sentence now subject, predicate and object are thoughts. The exchange of the triple (P1:B.OBAMA, P1:BIRTHPLACE, P1:HONOLULU) from person $p1$ to person $p2$ via the described symbolic indirection is visualized in Figure 2. Note that the thoughts of $p1$ and $p2$ are disjoint as they are in different brains, but hopefully they are similar enough, so that $p2$ understands $p1$. Also note that P2:B.O. is more detailed than P1:B.OBAMA, as $p2$ also knows a more specific name.

From this simple example one can identify two problems that can occur in our communication:

- A thought can be referred to by several symbols (e.g., *Barack Hussein Obama II*, *Obama*, *President of the*

[6] We focus on written or spoken communication here.

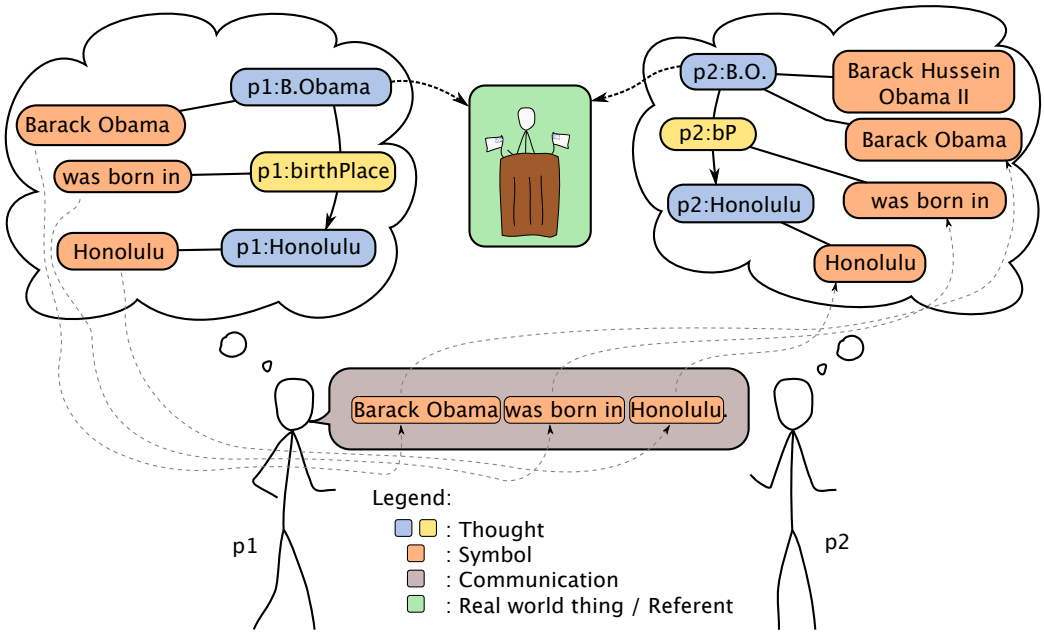


Figure 2: Human to human communication

United States, *President Obama* all could refer to one single thought P2:B.O.).

- A symbol can refer to several thoughts, also known as lexical ambiguity (*Apple* might refer to APPLE (the fruit) or APPLEINC (the company)).

3.1 Human associations

Now, when we read or hear some symbol (e.g., *Barack Obama*) we instantly connect it with its thought(s) (e.g., P1:B.OBAMA), but also, we somehow remember related thoughts (e.g., P1:PRESIDENTUSA, P1:USA, ..., P1:HONOLULU), called associations.

Associations can be thought of as a graph of thoughts or a so called spreading activation semantic network [Collins and Loftus, 1975], in which each thought is a node and associations between these nodes are edges. An example for such a semantic network is depicted in Figure 3 and shows some possible associations of P1:B.OBAMA. The length of an edge represents the strength of the association (e.g., P1:B.OBAMA's association to P1:PRESIDENTUSA is stronger than to P1:HONOLULU).

3.2 Context

It is argued that such associations are key requirements for our daily communication [Gerrig and Zimbardo, 2010, pp. 240ff]. Whenever we read or hear a symbol, our brain looks up its connected thought(s). These thoughts and their associations (in the following called associations of the corresponding symbol) are used to generate and adjust a *context* of thoughts that are related to the current communication sequence. The associations of a symbol may either support the current context or oppose it. In the first case our associations reinforce the context and our confidence that we understood what the speaker was telling us and that we are thinking about similar thoughts rises. In the latter case we have an indication that our lookup of the symbol's thought was wrong and we struggle to find a different meaning for what we have heard so far (e.g., in the sentence "Last year the pen was abandoned as it was too dirty for the animals to live in." [Gerrig and Zimbardo, 2010, p. 241] one would first assume that "pen" was a writing instrument, as it does most of the time (so called biased ambiguity), but when reading about animals living in it we suddenly realize that it refers to an enclosure).

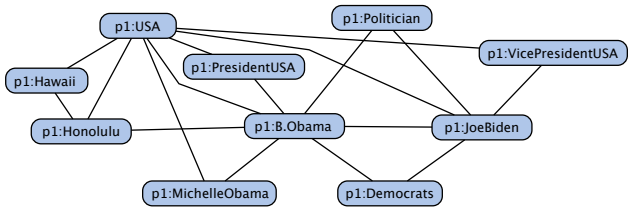


Figure 3: Semantic Network

Such a context, which emerged from our ongoing integration of associations, can be seen as our condensed set of thoughts related to the current communication. We can also think of this context as a working space, a drastically reduced search space for things that we expect to be related to what the speaker / writer expressed, instead of considering every thought we have in our mind to be potentially related. The context hence allows us to do much faster lookups (e.g., for symbols of thoughts which until now only were mentioned implicitly or for the integration of new information into our memory) and gives us the ability to resolve ambiguities.

4 Comparing Linked Data with human associations

Linked Data and spreading activation semantic networks used to model human associations are very similar. Both can be seen as network structures with thoughts as nodes and edges in between, both of huge dimensions.

In contrast to human communication, "thoughts" in Linked Data are meant to be exchanged directly between machines, which means without the symbolic indirection that we humans have to use. In contrast to the human brain the evolving knowledge base is thus distributed and interlinked, which is at the same time one of its largest advan-

tages and disadvantages. For example it allows continuing growth and allows machines to exchange information without reasoning about whether their symbols mean the same things, but at the same time this freedom introduced many errors into Linked Data, as anyone can issue statements about anything or as people use properties in different ways (cf. the use of `owl:sameAs` [Halpin and Hayes, 2010]).

Another difference between Linked Data and human associations is that Linked Data has strongly typed edges, which identify the kind of relation between two concepts very precisely, while human associations do not. Usually it takes us much longer to find a name for the association of two thoughts than it takes us to name the associated thought. Furthermore, human associations do not seem to be limited to a certain type of relation between thoughts, even though it would be interesting to investigate which of these properties we use most frequently.

To be able to investigate questions as the one above, and more general to perform any quantitative comparison between Linked Data and human associations, a large ground truth of human associations is needed, which currently does not exist.

Except from the need for such a ground truth, until now only differences were mentioned, which render Linked Data more specific than human associations and hence would allow to easily simulate human associations with Linked Data by just ignoring details such as edge labels. Still there exists one key feature of human associations, which currently is not part of Linked Data: association strengths. Even though heavily dependent on the person, the current context and task, nearly all humans will agree that they generally associate Barack Obama stronger with USA than with Honolulu. In contrast to this, the facts in the Linked Data cloud are facts in a logical sense. They are assertions, all of the same "truth", none being more valuable than another.

A collection of such association strengths would allow us to ask machines not only to give us all information about an instance (e.g., all 600 triples for Barack Obama), but also to rank this information by association strength (e.g., only presenting the top 10 of these to an end user) and thus, to constrain the number of results. With regard to understanding human texts, this would allow us to propagate activation from thoughts whose symbols directly occur in the text to thoughts occurring only implicitly in a human like way. This in turn, would enable us to narrow down our search space from the whole Linked Data graph to only those thoughts associated with the current communication sequence by an "average human". Another immediate benefit from annotating Linked Data triples with an association strength is a kind of feedback for automated extraction processes such as the one underlying DBpedia. One could investigate, which extraction rules yield high and which ones yield low association strengths, possibly driving an improvement process.

Besides these immediate uses, such a collection of association strengths would also allow us to test whether current common heuristics truly model how we associate thoughts and if they do, they could be used to bootstrap the acquisition of associations strengths. Examples for such heuristics include word co-occurrences on websites or graph intrinsic features such as page rank, betweenness and the like, trying to model how much activation flows from one thought to another.

For further research it is thus proposed to generate both: a human association ground truth as described earlier and a collection of Linked Data triples rated with their association strengths.

5 Linked Data Games

In order for both datasets to be of any use, their size needs to cover a sufficiently large amount of human associations / Linked Data. The larger the amount of data collected, the more meaningful later quantitative comparisons will be. At the same time all data collected (e.g., associations of Barack Obama or whether USA is stronger associated to him than Honolulu), is highly subjective, resulting in a large amount of input from different persons to be needed in order to arrive at a collectively agreed association / rating. Additionally entering a large amount of associations or ratings is an extremely tedious task.

The former constraints render the application of traditional approaches to generate a ground truth, such as paying test persons to record or write down their associations or ratings, infeasible. Nevertheless, there exists a novel approach called Human Computation [von Ahn and Dabbish, 2008] that might be suitable to solve this acquisition problem. The approach suggests to turn problems which are difficult to solve for machines into fun games with atomic decisions which are easily answerable for us humans.

In the following for each of the needed datasets an idea is presented on how the individual problem could be turned into such a game.

5.1 Building a human association ground truth

In order to acquire a human association ground truth one would usually present an item to a participant and ask her to enter all associations she has with this item in free text in a fixed amount of time (e.g., one minute). Then the next item would be shown. In order to make the collected data robust against priming effects, the order of the items usually would be randomized.

In order to turn this tedious process into a game many aspects of the ESP Game [von Ahn and Dabbish, 2004] can be borrowed. In contrast to the ESP Game the items of this game are not web images but Linked Data resources (i.e., their symbols).

In its simplest form the envisioned game called *Associator* is a symmetric two player output agreement web game, where a player starts a game and is randomly grouped with some other player for a short, fixed period of time (e.g., 2 minutes). Both players then play in rounds and can not communicate by other means than described in the following. At the start of each round they see an input item $\mathcal{S}$ (e.g., *Barack Obama*). What makes this game a Linked Data game is that this item is a symbol of a Linked Data resource $\mathcal{R}$ (i.e., there is a triple ($\mathcal{R}$, RDFS:LABEL, $\mathcal{S}$) in the LOD cloud (e.g. (DBPEDIA:BARACK_OBAMA, RDFS:LABEL, *Barack Obama*)). Both players are then asked to enter what their partner will associate with this item. The round will last until any of the outputs of player p_1 matches any of the outputs of player p_2 or until both players decide to pass this item, as they can not agree on any association. All outputs and timings in this process are recorded. In case of a match both players get points depending on the time it took them to find the match and get into the next round. In case of a pass both players get into the next round without getting points.

What this game is going to return behind the scenes is a collection of associations between the input items (i.e., Linked Data resources' symbols) and user entered symbols, which actually represent associated thoughts as we know. The matches recorded in this process are very good candidates for collectively agreed associations of the presented item. The other non matching guesses as well pose associations, but are very susceptible to subjective associations, cheating and the like. Still any recorded round can be used for single player games, for example when there is an unequal amount of players who want to play or when a player quits out of a running game. In this case the current player is not matched with another player who is playing at the same time, but one whose session was recorded earlier. This single player process can then validate additional guesses of the recorded session.

In contrast to the ESP Game a couple of open questions remain trying to use the collected data to build a human association ground truth.

First of all this simplest version of the envisioned game does not include taboo words. Taboo words are a list of words shown to the players in addition to the round's item. Any word in the taboo list can not be used as a match to get into the next round. In the ESP Game taboo words are used to force the players to enter a larger variety of labels, by adding outputs that are agreed on for a certain amount of times to the corresponding input's taboo list. The problem with this approach is that taboo words certainly bias the association process in a way that not only the round's input item primes associations, but also the taboo words (they sort of work as additional inputs). One of the ways to weaken this problem is randomly selecting taboo words out of an actually larger list of taboo words, as it is proposed for the ESP Game. Still it might be interesting to investigate if it is not possible to completely eliminate this kind of biasing, e.g., it is proposed here to try a kind of covered taboo list. In this case the players would only see a list of covered words, indicating that there are taboo words. In case the player enters a word from this list, it blocks a potential match, gets revealed (for this player only) and the player is awarded with some small amount of additional points.

Another open problem is related to the fact that all collected associations actually are symbols for associated thoughts. In order to analyze whether these associations are part of the LOD cloud already, the symbols need to be matched back to Linked Data resources, where possible. This process often is very ambiguous and it is not clear if this symbol to resource resolution can be seamlessly integrated into the Associator game without introducing new distortions. Also it should be investigated how to select the input items, which in the presented form are symbols of randomly selected Linked Data resources.

5.2 Rating Linked Data triples with association strengths

While the previous game idea targets creating a human association ground truth, in this section an idea shall be presented how to turn the process of assigning association strengths to Linked Data triples into a game. A simple traditional approach for this task could show all Linked Data triples related to a previously selected Linked Data resource (e.g., all triples with `dbpedia:Barack_Obama` as subject and/or object) to a test person and ask her to order these triples according to which one of the relations she would have thought of first, when hearing or reading *Barack Obama*. After ordering the list, the next list for another Linked Data resource is shown.

Before presenting the idea to turn this process into a game, two things shall be mentioned: First, as showing a list of URI triples to the end-user is not of much use, the users will always see a symbolic representation of this list. Luckily Linked Data resources are usually labeled with a symbol with the `rdfs:label` datatype property. Second, the outcome of each of these experiments, which is a user centric absolute ranking, is not only highly subjective, but sometimes even unstable for one person, as a lot of relative decisions are involved within this process and the human brain tends to lose track of them. Hence it seems to be better to ask for these atomic relative comparisons of two facts and then use an objective ranking algorithm to generate an absolute ranking out of them. Generating an absolute ranking out of such results can be compared to chess ranking systems, where based on the outcomes of atomic competitions (player p_1 won against p_2), a global ranking is calculated, just that in this case there are no players competing, but facts.

The envisioned game is called *BetterRelations* and borrows many ideas from Matchin [Hacker and von Ahn, 2009], which asks users to compare images against each other and calculates a global ranking from this.

In its simplest form BetterRelations is a symmetric two player decision agreement web game. A player starting the game is randomly matched with some other player either for a predefined amount of time (e.g., two minutes) or a predefined amount of rounds (e.g., 50). At the start of each round both players are presented with one input item, which actually is a Linked Data resource's symbol (e.g., *Barack Obama*), and two facts about this item (e.g., *is president of the USA* and *was born in Honolulu*). Both players are then asked to select the fact that their partner will have thought of first. In case both players agree they are rewarded with points and get into the next round. In case of disagreement, both players get into the next round but do not get points. As in Matchin the amount of points collected in a round rises with the number of decision agreements of both players in a row, punishing a disagreement without actually subtracting points from the user and preventing cheat strategies to gain many points by simply selecting a random fact as fast as possible. In order to prevent another obvious cheating strategy, the facts are presented to both players in randomized order. All actions in this process are recorded, and for example used for a single player mode.

Behind the scenes this game acquires a large amount of relative decisions between facts. Decisions which both players agreed on are especially validated. Disagreements could mean a lot of things and could result from subjective rankings and cheat attempts, or could indicate that the two facts were equally important.

In contrast to Matchin, BetterRelations will not create one globally ranked list for all Linked Data triples, but instead is going to create a list for each Linked Data resource of interest and all facts related to this resource. The ranking algorithm, which transforms the relative ratings into these global ratings hence has to deal with a lot smaller lists than in the case of Matchin.

Also the algorithm shall be able to quickly exclude a large number of erroneous facts, as they occur in Linked Data, from being played again, in order not to bore players with such facts. One possibility would be to provide

the players with an explicit button saying "both facts are nonsense". Nevertheless, it has to be investigated how to include this third choice into the rewarding system without abandoning the effective cheating prevention mechanisms.

Last but not least, BetterRelations is a game of a very exploratory nature. Often it could happen that the player does not recognize the round's item and might want to read a few sentences, explaining what the resource is about. For many Linked Data resources such a short introduction exists in the form of a `rdfs:comment` or a `dbpedia-owl:abstract`, but it is unclear how this information can be included in the game without interfering with the rating choices. One promising possibility could be to include such information on demand only and to treat the resulting rating in a different way on the server side. Also as it generally could take a few seconds for the user to switch context, it should be investigated if the throughput of the game can be raised by playing a few consecutive rounds about the same item, only changing the facts, without introducing priming effects.

6 Conclusion and Outlook

This paper introduced and motivated the hypothesis that simulating human associations could improve text understanding capabilities of machines. Thanks to Linked Data we have a very large and promising dataset at hand to simulate human associations. Nevertheless, in order to investigate more thoroughly whether Linked Data really is a good dataset to simulate associations, first of all a reasonably large human association ground truth is needed. Also as human associations can be of different strengths, such associations strengths would need to be annotated to many Linked Data triples.

As generating both of these datasets would be infeasible with traditional approaches, for each of them an idea was presented how to turn the acquisition process into a "Game with a Purpose".

After presenting these ideas the next research steps include implementing several versions of the games and properly evaluating them with respect to the desired data quality, the throughput, average lifetime play and expected contribution as mentioned in [von Ahn and Dabbish, 2008].

The resulting datasets are then to be used to investigate the overlap of human associations and Linked Data, to rate the extraction process of DBpedia and to benchmark several heuristics used to infer non-existent edge weights for Linked Data.

Acknowledgements Special thanks to Benjamin Adrian for the many insightful conversations about this field of research.

References

[Baddeley *et al.*, 2009] Alan Baddeley, Michael W. Eysenck, and Michael C. Anderson. *Memory*. Psychology Press, 2009.

[Berners-Lee *et al.*, 2001] Tim Berners-Lee, James Hendler, and Ora Lassila. The Semantic Web: A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities. *Scientific American*, 284(5):34–43, May 2001.

[Collins and Loftus, 1975] Allan M. Collins and Elizabeth F. Loftus. A Spreading-Activation Theory of Semantic Processing. *Psychological Review*, 82(6):407–428, 1975.

[Gerrig and Zimbardo, 2010] Richard J. Gerrig and Philip G. Zimbardo. *Psychology and Life*. Allyn & Bacon, Pearson, Boston, USA, 19th edition, 2010.

[Hacker and von Ahn, 2009] Severin Hacker and Luis von Ahn. Matchin: Eliciting User Preferences with an Online Game. In *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1207–1216, Boston, MA, USA, 2009. ACM.

[Halpin and Hayes, 2010] Harry Halpin and Patrick J. Hayes. When owl: sameAs isn't the Same: An Analysis of Identity Links on the Semantic Web. In Christian Bizer, Tom Heath, Tim Berners-Lee, and Michael Hausenblas, editors, *Proc. of the WWW2010 Workshop on Linked Data on the Web (LDOW)*, Raleigh, USA, 2010. CEUR Workshop Proceedings.

[Siorpaes and Hepp, 2007] Katharina Siorpaes and Martin Hepp. OntoGame: Towards Overcoming the Incentive Bottleneck in Ontology Building. In *Proc. of the 3rd International IFIP Workshop On Semantic Web & Web Semantics (SWWS)*, Vilamoura, Portugal, 2007. Springer, LNCS.

[von Ahn and Dabbish, 2004] Luis von Ahn and Laura Dabbish. Labeling Images with a Computer Game. In *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*, pages 319–326, Vienna, Austria, 2004. ACM New York, NY, USA.

[von Ahn and Dabbish, 2008] Luis von Ahn and Laura Dabbish. Designing games with a purpose. *Communications of the ACM*, 51(8):58–67, August 2008.

[von Ahn *et al.*, 2006] Luis von Ahn, Mihir Kedia, and Manuel Blum. Verbosity: A Game for Collecting Common-Sense Facts. In *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*, pages 75–78, Montréal, Québec, Canada, 2006. ACM, New York, NY, USA.

Reuse of Pharmaceutical Experience on Patient-individual Formulations

Mirjam Minor
University of Trier
D-54286, Trier, Germany
minor@uni-trier.de

Michael Raber
St. Barbara Apotheke
D-54290 Trier, Germany
apotheke-trier@arcor.de

Abstract

This paper presents on a novel knowledge and experience management approach for the development of patient-individual formulations in pharmacies applying case-based reasoning to solve problems with formulations that are under construction. A detailed analysis of the tasks and problems of formulating mixtures is given. A sophisticated domain model, a case base for problem solving and some first services are described that support pharmacists in their daily formulation tasks. The evaluation results from a first field test are presented. A brief discussion of related work and an outlook conclude the paper.

1. Introduction

Pharmacies are not only dispensing medications that have been fabricated by the pharmaceutical industry. They are also compounding medications for individual patients. The development of a formulation for an efficient, safe, and stable mixture requires plenty of task-oriented, pharmaceutical knowledge. A huge amount of this knowledge is documented in heterogeneous information sources like pharmaceutical textbooks and electronic publications[1]. It is a difficult and time-consuming task to check all relevant properties of a new mixture. Even experienced pharmacists have to review the literature carefully for relevant information, as there is such a wide variety of prescriptions and prefabricated substances. In current pharmacy practice, a profitable creation of patient-individual formulations is hardly possible. Task-oriented, electronic assistance is urgently needed.

This paper presents a knowledge and experience management approach [Bergmann, 2002] that supports pharmacists in formulating drugs for individual patients by means of a pharmaceutical domain model and case-based reasoning. It focuses on *modeling* pharmaceutical knowledge on substances, properties, and relationships of them, *using* the properties and relationships as check criteria in order to examine formulations for efficacy, safety, and stability, and for providing drug information, and employing case-based reasoning for *reusing* cases that record solutions of hurt check criteria.

The paper is organized as follows: In Section 2, we introduce the tasks and problems of patient-individual formulation. Section 3 addresses our knowledge and experience management approach in order to support the formulation tasks. Section 4 presents first evaluation results from a field test. Section 5 concludes the paper with a discussion and an outlook.

2. Development of a patient-individual formulation

A pharmacist develops a patient-individual formulation usually on the basis of a medical *prescription*. The prescription form contains a list of quantified substances called *prescription positions*. Figure 1 shows a sample prescription with a prefabricated salve 'Volon A' in the first position and a basic skin cream 'Basiscreme DAC'[2] in the second position. The 'aut idem' label on the left hand side means that the substance may be substituted by a generic drug.

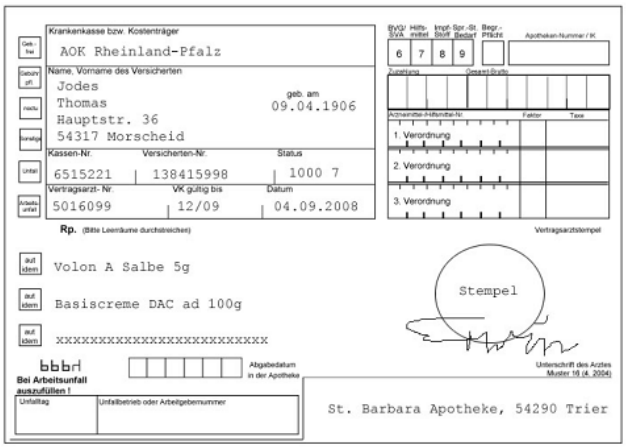


Fig. 1: Sample prescription form handed in at a pharmacy.

The pharmacist extracts the relevant information from the prescription in order to acquire an *initial formulation*. The initial formulation has to be checked carefully and to be adapted if necessary. With the availability of the *final formulation*, the expiration date and further drug information can be determined.

[1] A prominent German sample for an electronic source of over 500 formulation-related references is the formularium NRF ('Neues Rezeptur-Formularium') of the ABDA ('Bundesvereinigung Deutscher Apothekerverbände'), online available at http://www.pharmazeutische-zeitung.de/index.php?id=2264, Retrieved October 24, 2008.

[2] DAC stands for the German drug codex '**D**eutscher **A**rzneimittel **C**odex', which specifies the components of some basic pharmaceutical substances. The DAC is part of ABDA's NRF (see previous footnote).

2.1 Creating an initial formulation from a prescription

The first step of drug formulation is to create the initial formulation from the prescription. Several prescription-related challenges have to be addressed concerning the substances to be processed:

1. Synonymous names for substances,
2. Heterogeneous units of measurement,
3. Prefabricated substances.

ad 1: Substances can be named by a variety of synonymous names. For instance, the common dermatological agent 'urea' is often prescribed by its Latin form 'carbamidum', or by 'carbamide', 'carbonyl diamide', or 'carbonyl diamine'. Sometimes, also trade marks like 'Aspirin' instead of 'acetylsalicyl acid' are taken. A pharmacist is hardly able to know all the names for substances that may be used by the physicians. In practice, she successively learns the preferences of particular physicians only in addition to her own pharmaceutical language use. Novel names have to be looked up when they occur.

ad 2: The second difficulty is the variety of measurements in which the substances can be denoted. Mass values can be seen alongside volume and proportion specifications.3 A missing unit means that the value is interpreted in gram. The supplementary 'ad' stands for filling up the whole mixture with this substance until the specified amount. For instance, the 'ad 100 g' in the above sample prescription specifies that the pharmacist should add an amount of 95 g of this basic cream. Frequently, the prescription positions are given in different units. That means that the units of measurement have to be transformed into unified, explicit specifications.

ad 3: Positions with an explicated unit of measurement may still suffer from incomplete knowledge concerning the content. This is to be found if a compound substance is denoted as it is frequently the case with prefabricated substances. Sometimes, detailed information on the particular ingredients is missing. 'Volon A', for instance, contains 0.01 g Triamcinolon per gram, i.e. 0.05 g in the whole sample mixture above. Furthermore, the vendor's specification of 'Volon A' says only that it contains Polyethylen and Paraffin. The proportion of these two auxiliary substances is not published by the vendor. The pharmacists have to estimate the amounts of subcomponents if necessary. The estimation can be feasible over multiple assembly levels.

When the pharmacist has solved the prescription-related problems that we described above, the initial formulation is available. Before the mixture is actually made, the initial formulation has to be reviewed for efficacy, safety, and stability.

2.2 Validating a formulation and solving problems

The review of an initial formulation requires pharmaceutical knowledge on the basic and prefabricated substances, their properties and relationships, as well as on the particular patient, and use. Obviously, the concentration of active components is very important, but also potential problems with particular substances and mixtures have to be considered. The check criteria for validating a formulation can be assigned to the following three areas:

I. Physical, chemical, and galenic characteristics of a certain substance or a group of substances,
II. Physical, chemical, and galenic characteristics of a combination of certain components,
III. Microbiological quality of a mixture.

ad I: First, the pharmacist confirms that the particular substances are medically unobjectionable for a certain patient and use. Phenol, for instance, is not any more recommended to be applied on the skin. Furthermore, she checks whether the concentrations of the active components of the drug, i.e. of the medical and preservative agents, are within the mandatory range. Further characteristics to be validated are, for instance, the range of pH-values in which a substance should be formulated best or which impact a type of wrapping has on the stability of a particular substance.

ad II: Second, the pharmacist clarifies that there are not any serious incompatibilities of components of the formulation. This concerns, for instance, the physical stability of two components in a mixture. The pharmacist knows whether an agent is soluble sufficiently in an auxiliary substance like olive oil or whether problems like sedimentation or caking are to be expected.

ad III: The microbiological quality of a mixture is not only depending on the use of preservative substances. It also plays a role to avoid the generation of a nutrient medium within the mixture by a certain combination or treatment of substances. Sometimes, the decisions of the pharmacist rely on incomplete knowledge for the reason of incomplete specifications of substances that we have described above or for other reasons.

If the pharmacist detects a problem during the validation of a formulation it can be solved in different ways. Sometimes the pharmacist is able to mend the potential problem without changing the formulation, for instance with a soon expiration date or with adding the instruction that the drug has to be refrigerated. If a major adaptation of the formulation, not such as the addition of a preservative agent, becomes necessary the prescribing physician has to be contacted in order to develop a solution conjointly. A sample adaptation that requires the approval of the physician is depicted in Figure 2. Sometimes, the physician decides not to adapt the formulation, for instance for the reason of tradition or for intentionally prescribing a placebo that is effective psychologically only.

[3] The ratio '% (m/m)' stands for 'mass per mass', for instance '% (g/g)' for 'gram per gram'. '% (m/V)' is for 'mass per volume' like '% (g/ml)'. '% (V/V)' units for 'volume per volume' are quite common as well as mass proportions with respect to drops, pieces, or international units.

Uncommittedly composed formulation

Acid. salicyl. plv.	5.0
Triamcinolonacetonid	0.1
Ol. oliv.	ad 100.0

Problem

- solubility of Salicyl acid in olive oil 2.5 % only; consequence: sedimentation, growth of crystals
- Triamcinolonacetonid not solvable in olive oil, consequence: sedimentation, caking
- restricted temperature exposure for vegetable oils
- solubility by shaking the drug is not sufficient

Alternative

Salicyl acid	5.0
Triamcinolonacetonid	0.1
2-Propanol	10.0
Octyldodecanol	ad 100.0

Directions: solve Salicyl acid in Octyldodecanol while heating and separately Triamcinolonacetonid in 2-Propanol. Let both solutions cool down and mix them at room temperature.

Fig. 2: Sample adaptation of a formulation [RLH03, own translation].

3. Knowledge and experience management approach

The analysis of the development of patient-individual formulations in Section 2 has shown that it would be worthwhile to have an assistant system for the management of the required pharmaceutical knowledge and experience. In the following, we will describe the modeling and reuse of pharmaceutical knowledge in a knowledge and experience management system. The system supports the following tasks of formulation:

- Acquisition of initial formulation,
- Retrieval of relevant check criteria,
- Review of check criteria,
- Mending of hurt criteria (deactivate criterion, adapt formulation: re-calibrate, change list of positions),
- Generation of additional instructions for how to prepare and use a mixture.

The degree of automatic support that is provided by the system varies from task to task: The acquisition of an initial formulation from a prescription form is done mainly automatically but requires user interaction for the acquisition of missing information on ingredients and amounts within prefabricated substances (compare Section 2.1). The retrieval and review of the check criteria is performed automatically. It addresses the validation areas I – III that have been introduced in Section 2.2. The mending of hurt criteria is supported by a case-based approach. The generation of instructions has still to be done by the pharmacist.

In order to realize the support capabilities mentioned above, the assistant system provides a set of services using an underlying domain model. Section 3.1 will introduce this domain model. Section 3.2 will briefly sketch the services.

3.1 Domain model

The domain model represents knowledge on patient-individual formulations in a task-oriented way. That means that it does not aim to describe general knowledge on substances like all chemical, physical and galenic properties of substances. Instead, it focuses only on those characteristics that are relevant for the tasks of formulation. The model consists of four main parts:

- *substances*: a task-oriented taxonomy of substances, their properties, and relationships,
- *formulations*: data on prescriptions and formulations including the routes of application and the wrappings,
- *master data* of patients, health professionals, and health insurances, including administrative as well as medical information like history data on a patient's prescriptions,
- *system administration data* like user roles.

In the following, we will present some details on the substances as this part of the model has the most important impact on providing assistance for the formulation tasks.

The core of the substances model is a taxonomy of substances, groups of substances, and prefabricated substances. Figure 3 depicts the section of this taxonomy for the prefabricated substance 'Volon A'. It consists of three ingredients: the medical agent Triamcinolon with 0.01 % (g/g), Polyethylen, and Paraffin, which belongs to the group of Alkanes.

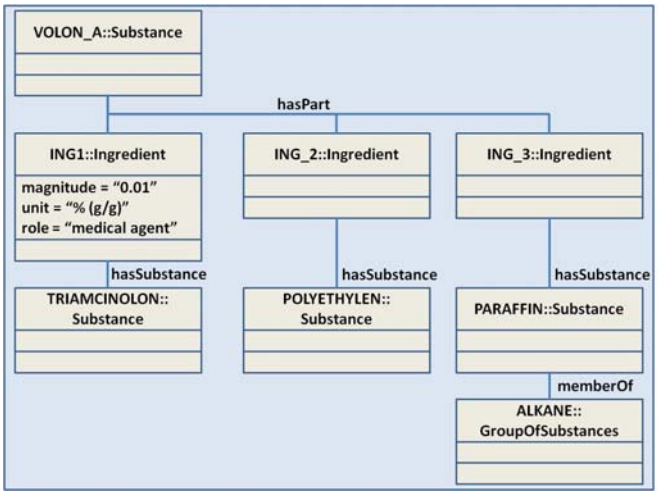


Fig. 3: Sample section of the substances model describing Volon A.

Synonym names are not depicted in Figure 3 due to space limitations. They are integrated as objects of the class Synonym that are related to the abstract term, for instance 'Volon A::Synonym' and 'VolonA::Synonym' related to 'VOLON_A::Substance'.

Check criteria are represented by objects of the class 'CheckCriterion' encapsulating terms in predicate logics. For instance, the term `Phenolic(x) ∧ (Cyclohexene(y) ∨ Benzol (y))` stands for the following sentence describing an incompatibility: "Phenolic substances might react with Benzol or Cyclohexenes producing unwanted phenolic by-products". Atomic formulas with the arity one (one predicate about one term) stand for

check criteria on properties of single substances within a mixture like '`AppropriateConcentration(x)`'.

3.2 Services

The domain model that we described in the previous subsection is used by four services at the moment: a service *build data model*, a *formulation service*, a service *maintain system administration and master data*, and a service *maintain substances*. In the following, we will have a closer look at the formulation service, which is called when a new formulation is to be developed. First, the acquisition of the initial formulation has to be done, and then the validation and adaptation of the formulation takes place.

The acquisition starts with building a formulation object and performing a depth first search on the positions of the prescription object to create new position objects for the formulation object. The already existing position objects of the prescription are copied and not overwritten for reasons of documentation. The result is a tree of all substances that are involved in the formulation. Then, the specification of quantities is completed and unified. After the choice of either mass or volume specification, a breadth first search through the tree of substances is executed in order to unify the units of measurement and to complete the missing units. If quantities cannot be derived automatically, the pharmacist is involved for interactively estimating quantity values. The order of dealing with the quantity values is important: In each node of the tree, the mass and volume specifications are handled before expanding the proportion specifications. A special treatment is required if an 'ad' occurs. First, the overall quantity of the mixture is determined from the 'ad' position, second, the quantities of the other positions is computed according to the order mentioned above, and last, the quantity of the 'ad' position is derived.

The validation consists of the retrieval and review of relevant check criteria. It begins with computing the concentration of all substances and storing them in a list. Then, the relevant check criteria are retrieved and tested automatically. In the prototypical implementation, a linear search is employed as retrieval method. This impacts a high computational complexity, which is to be optimized in future.

The adaptation of the formulation when a hurt criterion has been detected is supported by case-based reasoning [Richter, 1998]. A *case* consists of a hurt criterion in any formulation (the problem part) and an alternative set of substances or additional preparation directions (the solution part). In Figure 2, the uncommitted composed formulation together with the four problem items form the problem part of a case while the alternative positions and the directions describe the solution part. A *case base* consists of a set of such cases recording pharmaceutical experience. If a *new problem* occurs that requires the adaptation of a formulation, the best matching case is retrieved from the case base in order to reuse its solution. The retrieval is based on a standard similarity function [Bergmann, 2002] computing a weighted sum of local similarity values based on an internal, structural representation of the problem part of the cases. When building the internal representation of the cases, the domain model is applied to create attribute-value pairs from the initial formulation positions. The representation is created by the same preprocessing algorithm that is applied during the processing of a new recipe for the acquisition of an initial formulation. The user decides after the retrieval of the best matching case whether the solution is applicable to the new problem and transfers it to the new formulation where appropriate.

4. State of implementation and evaluation

The acquisition of initial formulation, the retrieval, and the review of check criteria is fully implemented. The mending of the hurt criteria by means of case-based reasoning is ongoing work.

A field test has been conducted with three German pharmacies to evaluate the already implemented parts of the approach. As all German pharmacies underlie the same accounting mechanism with the health insurance companies and the major part of them makes use of the electronic sources described in Section 2, the evaluation results can be considered representative for Germany. The field test investigated two research questions: The first question is whether the domain model including the check criteria is appropriate for the pharmaceutical knowledge that is required for the task of creating patient-individual formulations. The second question is whether the implementation is usable. The first question has been investigated by an expert review of the domain model. The second question has been investigated by one of the pharmacists involved by means of working with the implemented modeling tool. The results for the first question are quite promising: The model review has shown that it covers the pharmaceutical knowledge including the check criteria to a great extend. Only the potentially heterogeneous granularity of check criteria was considered to cause problems in future. The modeling activities concerning the second research question led to 61 substances and groups of substances, 18 check criteria derived from 11 genuine recipes. It turned out that the tool worked in principle well but that the computational performance of the tool should be improved in future.

5. Discussion of related work and outlook

In this paper, we have presented a knowledge and experience management approach that supports pharmacists in elaborating patient-individual formulations. A domain model of substances including their properties and relationships is used to check potentially problematic properties and relationships for a certain mixture. A case-based approach provides assistance for the adaptation of formulations when check criteria have been hurt.

The literature reports a case-based approach using decision trees to guide tablet formulation [CWR98]. In contrast to our work, this approach addresses the formulation for the industrial production of drugs. In our patient-individual approach, aspects like tablet weight and yield pressure do not play any role. Furthermore, there is a wider variety of prescriptions in pharmacies than in an industrial tablet production. We think that considering particular check criteria is more feasible for this application area than using complex decision trees also with respect to maintenance issues.

In our future work, we aim at finishing the implementation and evaluating the case-based support for the validation and adaptation process.

Acknowledgments

The authors acknowledge the contributions of Thomas Jodes, Christian Schlimbach, Stefan Schölzel, and Michael Christmann who made valuable contributions to the project by preparing their diploma theses and implementing the prototype.

References

[Bergmann, 2002] Bergmann, R.: *Experience Management*. LNAI 2432, Springer, 2002.

[Craw *et al.*, 1998] Craw, S.; Wiratunga, N.; Rowe, R.: Case-based design for tablet formulation. In *Advances in Case-Based Reasoning, Proceedings of the 4th European Workshop on Case-Based Reasoning*, LNAI 1488, Springer, pp. 358-369, 1998.

[Eifler-Bollen *et al.*, 2003] Eifler-Bollen, R.; Lein, A.; Reimann, H.: Qualität von Rezepturen steuern. Retrieved October 30, 2008 from the Web http://www.pharmazeutische-zeitung.de/fileadmin/pza/2003-46/titel.htm.

[Richter, 1998] Richter, M. M.: Introduction. In (Lenz, M.; Bartsch-Spörl, B., Burkhard, H.-D., Wess, S., Eds.): Case-Based Reasoning Technology. LNAI 1400, Springer, pp. 1-15, 1998.

Resubmission: Taking OWL to Athens: Semantic Web Technology takes Ancient Greek History to Students

Jochen Reutelshoefer[1], **Florian Lemmerich**[1], **Joachim Baumeister**[1], **Jorit Wintjes**[2], **Lorenz Haas**[2]
1. Institute of Computer Science, University of Würzburg, Germany
{lastname}@informatik.uni-wuerzburg.de
2. Institute of Ancient History, University of Würzburg, Germany
{firstname.lastname}@uni-wuerzburg.de

Abstract

The HermesWiki project is a semantic wiki application on Ancient Greek History. As an e-learning platform, it aims at providing students effective access to concise and reliable domain knowledge, that is especially important for exam preparation. In this paper, we show how semantic technologies introduce new methods of learning by supporting teachers in the creation of contents and students in the personalized identification of required knowledge. Therefore, we give an overview of the project and characterize the semi-formalized content. Additionally, we present several use cases and describe the semantic web techniques that are used to support the application. Furthermore, we report on the user experiences regarding the usefulness and applicability of semantic technologies in this context.

1 Introduction

Students today are used to collect a large amount of required knowledge from the internet. While the number of available webpages is huge for almost any given topic, the quality of these pages is quite heterogeneous. Additionally, it is very difficult for students, especially undergraduate students, to extract the essential knowledge from extensive webpages. Therefore, students can significantly benefit from a reliable and concise knowledge base, that combines the advantages of traditional text books with the advantages of internet-based knowledge platforms, like accessibility, integration of multimedia resources, and the ability to update the contents easily. However, the creation of such a web application still is a tedious task, as texts and resources must be created, collected, and adapted to the context. Additionally, with the growing amount of content it again gets more difficult for students to find exactly the resources they require. In this paper, we describe how semantic technologies help to alleviate both problems in an e-learning project in the domain of Ancient Greek History, the HermesWiki.

The HermesWiki is a semantic wiki, developed to give students of history an introductory domain overview. As a full featured wiki system, HermesWiki allows for an effective collaborative composition of content. Furthermore, semantic technologies provide advanced functionality that generates additional value to the learning environment: Formalized paragraphs and semantic annotations of the main text enable the contributors to easily generate additional content tailored to different tasks, such as tables of events, maps, or a quiz. Further, students have access to different forms of knowledge like narrative elements, ancient source texts, geo data, chronologic information and images. They can use semantic navigation to easily find the concepts they are looking for, filter the presented content by its relevancy on demand, and query the systems knowledge using an intuitive user interface. Figure 1 shows the essay about *Tyrannis*. It includes a short list of the most important events generated from the ontology, an introductory section, that explains the context of this essay, a text description of the respective topic, an overview of ancient historical sources, and a formalized set of relevant events.

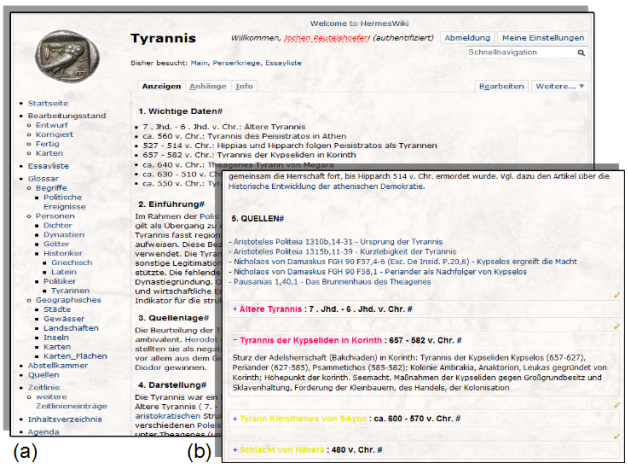


Figure 1: The top (a) and bottom (b) of the wiki page containing the essay about *Tyrannis* taken from the HermesWiki. (Screenshots of the system, in German language)

2 Resubmission

This is a resubmission of work that has originally been submitted to the 7th Extended Semantic Web Conference (ESWC2010) [Reutelshoefer *et al.*, 2010].

References

[Reutelshoefer *et al.*, 2010] Jochen Reutelshoefer, Florian Lemmerich, Joachim Baumeister, Jorit Wintjes, and Lorenz Haas. Taking OWL to Athens – Semantic Web Technology takes Ancient Greek History to Students. In *ESWC'10: Proceedings of the 7th Extended Semantic Web Conference*. Springer, 2010.

INTEGRATION OF LINKED OPEN DATA IN CASE-BASED REASONING SYSTEMS

Christian Severin Sauer[1] **Kerstin Bach**[2] **Klaus-Dieter Althoff**[2]

University of Hildesheim, Dep. of Computer Science
Intelligent Information Systems Lab
D-31141, Hildesheim, Germany
[1]christiansauer@gmail.com, [2]{lastname}@iis.uni-hildesheim.de

Abstract

This paper discusses the opportunities of integrating Linked Open Data (LOD) resources into Case-Based Reasoning (CBR) systems. Upon the application domain travel medicine, we will exemplify how LOD can be used to fill three out of four knowledge containers a CBR system is based on. The paper also presents the applied techniques for the realization and demonstrates the performance gain of knowledge acquisition by the use of LOD.

class	instances
Place	413,000
Person	312,000
Work	320,000
Species	146,000
Organisation	140,000
Building	33,000
...	...
Diseases	4,600

Table 1: Instances of DBpedia

1 Introduction

The recent developments in the field of Linked Open Data (LOD) further added value to the already existing vast amount of structured information that is available through the use of LOD [Bizer *et al.*, 2008]. The availability of such an amount of structured information suggests applying LOD to the semi-automatic generation of knowledge. In the context of our work knowledge represents the data that is organized in the knowledge containers of a Case-Based Reasoning (CBR) system. Each CBR system uses the four knowledge containers *vocabulary*, *similarity measures*, *transformational knowledge* and *cases*. The work presented in this paper focuses on filling the vocabulary and similarity measure containers as well as generating cases from LOD. Therefor, we demonstrate some possibilities of filling the aforementioned knowledge containers using simplified examples. These examples are preliminary, thus cannot be used in a real-world CBR system, because they are incomplete.

The linked data is provided for free and contains comprehensive knowledge where each term is assigned to at least one category and thus being represented in a specific context. The latest development in the field of LOD can be seen in the development of such complex knowledge repositories as for example given by the DBpedia ontology[1]. DBpedia contains many terms originating from the on-line encyclopedia Wikipedia. These terms are organized in an ontology and are being enriched with further information such as different labels in various languages. Currently, the DBpedia ontology contains 1,478,000 instances, i.e. Table 1.

Knowledge acquisition is still the bottleneck within the development of CBR systems. Our approach aims at automatization and for this purpose we propose a schema that extracts knowledge from LOD sources and provides and transforms it for CBR systems.

[1] http://dbpedia.org/ontology/

The remaining sections of this paper are structured as follows: In section 2 we describe the application domain travel medicine, especially the docQuery project in which the experiments are carried out, followed by section 3 that describes how knowledge can be integrated in CBR systems. Section 4 presents the implementation details of our approach. The following section 5 presents experimental results of our approach before we sum up the paper and give an outlook on future work in the final section 7.

2 Application Domain: Travel Medicine

Travel medicine is an interdisciplinary specialty concerned with the prevention, management and research of health problems associated with travel, and covers all medical aspects a traveler has to take care of before, during and after a journey. For that reason it covers many medical areas and combines them with further information about the destination, the activities planned and additional conditions which also have to be considered when giving medical advice to a traveler. Travel medicine starts when a person moves from one place to another by any mode of transportation and stops after returning home without diseases or infections.

The realization of the travel medicine project docQuery is based on the SEASALT (Sharing Experience using an Agent-based System Architecture LayouT) architecture [Reichle *et al.*, 2009] that is especially suited for the acquisition, handling and provision of experiential knowledge as it is provided by communities of practice and represented within Web 2.0 platforms. It is based on the Collaborative Multi-Expert-Systems (CoMES), approach presented by Althoff et. al. [Althoff *et al.*, 2007], a continuation of combining established artificial intelligence techniques and the application of the product line concept (known from software engineering) creating knowledge lines.

According to the SEASALT architecture, docQuery consists of eight different CBR systems. Each CBR system within docQuery equips a software agent that represents a

certain topic. Further the multi-agent-system is aggregating a composed result. However, within this paper we only focus on one CBR systems that contains information about diseases that might somebody can get infected with during a journey.

3 CBR System

CBR, the episodical knowledge, especially made experiences and successfully applied solutions for problems draw a huge potential for the development of an Experience Web [Plaza, 2008], because the experiences of many WWW users can be captured in case bases and provided in CBR systems. Because of the fact that CBR systems are based on experiences it is much easier to use them to handle experiences in comparison to systems that are based on technologies from the areas of information retrieval or semantic web. Experiences in CBR systems occur in all types of knowledge: mostly in cases, but also similarity measures, vocabulary and transformational knowledge can be derived. Huge amounts of raw data for the knowledge extraction is available on the web communities and the challenge is making these experiences available.

According to Richter [Richter, 1998], the knowledge of CBR systems can be provided in all four knowledge containers that include vocabulary, similarity measures, transformational (or adaptation) knowledge and cases. We focus on how the knowledge containers can be filled using the experiences provided in web communities. Therefore we have to extract the knowledge before it can be stored.

Further on, we deal with data sources that are mostly free text what usually requires a symbolic representation of keywords. That is the reason why we currently focus on the extraction of taxonomies that can be used for both, enhancing the vocabulary and assigning the similarity.

4 Implementation

Within this paper we present how knowledge about diseases can be extracted from LOD and integrated in a CBR system. Our CBR system has been developed using the open source tool myCBR [Stahl and Roth-Berghofer, 2008], thus all results produced by our application are compatible with the myCBR case representation and thus can be applied in myCBR-based applications. In case of the generation of taxonomies that contains knowledge about the similarity between objects represented in the taxonomy, we use the Knowledge Extraction Workbench (KEWo) to further refine the initially provided data from LOD. A more detailed description of KEWo can be found in [Bach *et al.*, 2010b] and [Sauer, 2010].

As the first step we retrieved data about diseases from the DBpedia ontology. The result of such a retrieval can be seen in figure 1. In this case we queried for all available diseases in the ontology and their German labels. The figure shows the shortened result containing the URI and the according label.

Furthermore, we queried the DBpedia ontology in order to extend our information retrieval by using the Resource Description Framework Schema (RDFS) to retrieve labels of diseases in different languages. Also, we retrieved the Simple Knowledge Organization System based (SKOS) information about the categories a disease belongs to, e.g. that a "bite" is a member of the category injuries. Analogous to the retrieval of German labels of different diseases we also retrieved the according German labels of these

Name	label of the disease
Viral_disease	true or false
Infectious_disease	true or false
Bacterial_disease	true or false
Injury	true or false
Foodborne_illness	true or false
Inflammation	true or false
Parasitic_disease	true or false

Table 2: Disease categories (derived from DBPedia)

categories. The technique used for these initial retrieval steps was a set of SPARQL queries conducted via the open source Desktop-SPARQL-Query tool "Twinkle"[2]. Listing 4 shows an example for a query retrieving all diseases from the DBpedia ontology together with the categories the are part of filtered in that way, that only the German labels of the diseases and the categories are returned.

Listing 1: Example Query

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>

SELECT ?disease ?categories ?dislabels ?catlabels WHERE
      {
?disease a <http://dbpedia.org/ontology/Disease> .
?disease rdfs:label ?dislabels .
?disease skos:subject ?categories .
FILTER (lang(?dislabels) = "de")
?categories rdfs:label ?catlabels .
FILTER (lang(?catlabels) = "de")
}
```

This still simple query, compared to the many sophisticated possibilities SPARQL offers to retrieve complex informations about objects and their relations, yielded 2004 category-disease-pairs with their corresponding German labels. Also some of the Items in the disease ontology are of a broader nature like the item Abuse we found enough specific Data to try to incorporate the retrieved Data into our CBR-system.

The simplest task was the extraction of data for the knowledge container vocabulary. For this task we simply derived the labels of the diseases from the retrieved data and thus formed a basic vocabulary of diseases for our CBR system. This vocabulary contains 2000 diseases with their German labels.

The next knowledge container we wanted to fill with Data from the LOD retrieval were the structured cases of our structured CBR system. For this purpose we chose six Categories from the English category-disease-pairs we assumed to be of interest for a case as attributes. The attributes that are describing a disease case are the following categories: *Viral_disease*, *Infectious_disease*, *Bacterial_disease*, *Injury*, *Foodborne_illness*, *Inflammation*, and *Parasitic_disease*.

Table 2 describes the structure of a disease case. The cases in our application scenario for this paper that are created based on LOD, of course, cannot serve as full case in a real-life application domain (like docQuery). However, LOD provides the potential data to create such cases.

To derive such cases from the retrieved data we implemented a java-based tool to extract the disease and category labels from the data and check if a disease label was part of a category-disease-pairs in which the category was, or was not one of the categories we chose to be the attributes

[2] http://www.ldodds.com/projects/twinkle/

disease	deutsch
<http://dbpedia.org/resource/AIDS>	"AIDS"@de
<http://dbpedia.org/resource/Acne_vulgaris>	"Akne"@de
<http://dbpedia.org/resource/Ankylosing_spondylitis>	"Spondylitis ankylosans"@de
<http://dbpedia.org/resource/Atherosclerosis>	"Arteriosklerose"@de
<http://dbpedia.org/resource/Decompression_sickness>	"Dekompressionskrankheit"@de

Figure 1: DBpedia retrieval result for disease (incomplete)

of our disease cases. After this extraction and comparison the resulting data was transformed into the csv-format to enable it to be imported as cases by the myCBR plugin we used in our Protégé development environment[3]. The created csv-file then was imported into Protégé resulting into the generation of 612 disease cases.

The third knowledge container we aimed at was the similarity measure. In our current project docQuery [Bach *et al.*, 2010a] we use taxonomies of items e.g. diseases, locations or medicaments to encode the similarity of these items into the distance of two items within the taxonomy [Bergmann, 2002]. So our goal was it to construct the according taxonomy from the retrieved LOD. This taxonomy, representing the similarity of diseases, was built using again a small java-based tool we implemented and one of our recently developed tools known as the Knowledge Extraction Workbench (KEWo). The process of taxonomy generation was thus the following: We extracted the diseases and the categories to which the diseases belong to as described above by deriving this information via a simple Java tool from the raw retrieval data. This process was again carried out on the German data set we retrieved. The resulting data was, due to the techniques used for taxonomy generation by the KEWo, further formated in a special way. The special formating listed the category twice followed by the disease. This resulted into a line describing a category-disease-pair looking as follows: *Mykose* (category) – *Mykose* (category) – *Kokzidioidomykose*(disease). This special formating is owed to the numerical approach the analysis methods of the KEWo employ. More details on the analysis with a brief discussion can be found in [Bach *et al.*, 2010b].

Our tool KEWo was created as an knowledge extraction tool operating on an on-line forum of travel medicine experts. Its goal is, in short, to derive medical knowledge from this forum employing various NLP and Information Extraction techniques on the content of the forum. For further details regarding the exact process model and principle of operation of the KEWo please see [Bach *et al.*, 2010b].

As a workaround to employ our KEWo for taxonomy generation from the retrieved LOD, the generated text,containing all category-disease-pairs was then inserted in the expert forum connected to the KEWo as a posting. Thus the KEWo was able to process the text containing the information from the LOD and so build a taxonomy of diseases.

We took the first 1000 category-disease-pairs and processed them in the described way, receiving a taxonomy describing 116 diseases. Figure 2 shows a snippet from the generated taxonomy. Such a taxonomy can be used in a CBR system as both, a source for vocabulary and the assignment of a similarity value between two concepts.

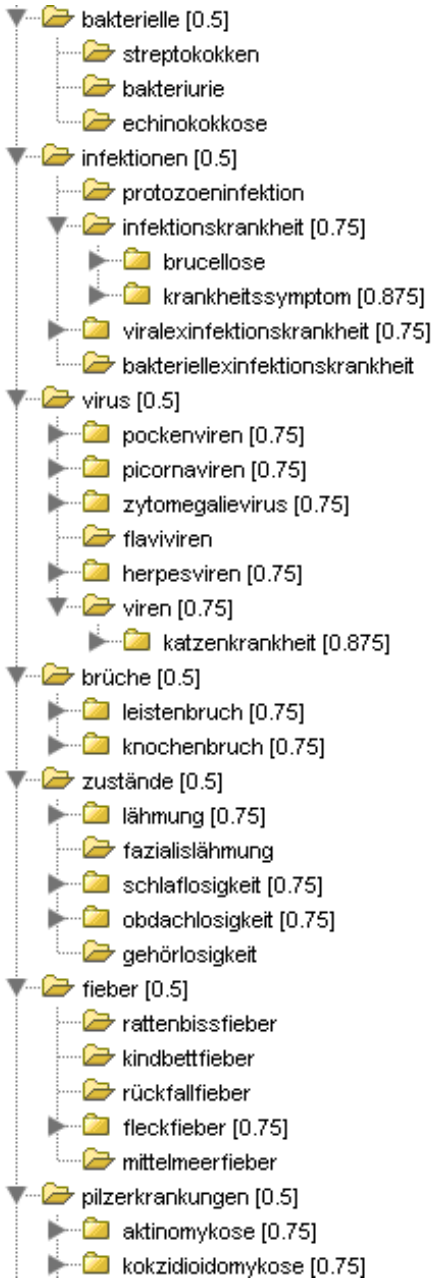


Figure 2: Snippet of the generated taxonomy

For the generation of the taxonomy we used the retrieved data about diseases and the according categories and con-

[3] http://protege.stanford.edu

ducted a statistical analysis regarding the occurrences of the different diseases and categories. In a later development it is, of course, desirable to integrate ontological data available in LOD directly into a similarity measure. However, our approach at this point is of a more basic nature in order to point out the possibility of simple data integration from LOD into a similarity measure.

It is obvious to ask the question why there are not more diseases in the taxonomy, given the fact that we used 1000 category-disease-pairs. The comparatively low amount of diseases showing up in the generated taxonomy is partly caused by a certain kind of 'missuse' of our own Tool KEWo which is optimized for analyzing natural language and not such highly structured text as given to it in this experiment, please see Section 5 [ref Section 5] for further details on this 'conflict'. Nevertheless we were in fact able to produce a taxonomy of diseases from the retrieved LOD and in the upper ranges of its structure the linking of the disease items are of satisfying quality.

5 Experimental Results

The goal of our experiments was it to retrieve LOD about diseases and incorporate the retrieved data into 3 out of the 4 knowledge-containers of a CBR-system. These three knowledge containers were the vocabulary, the cases and the similarity measure of a CBR system used in our project docQuery which aims for on-line knowledge provision in the field of travel medicine.

Our SPARQL query to the DBpedia ontology returned 2004 unique English disease-labels. Further queries returned 2000 German disease-labels, 2000 English category-disease-pairs in which a category-disease-pair represents a disease-label and a category it is assigned to in SKOS categories, e.g. *AIDS : viral_diseases* and also 2004 German category-disease-pairs.

For the first knowledge container, vocabulary we were able to extract all of the either English or German disease labels resulting into disease vocabularies consisting of 2004 English respectively German terms for diseases. We conclude that, the small initial amount of time invested into the development of the Java tool for the extraction of terms from the raw retrieved LOD aside, the amount of time for building a vocabulary is drastically reduced, literally to minutes. Once a tool is provided it is a matter of minutes to adapt the tool, run a new query, and extract the resulting terms from the retrieved LOD for any new vocabulary.

For the second knowledge-container we were able to extract 612 cases out of the English data retrieved. Each case consisted of a unique disease-name and six categories of diseases it either belonged to or not, please see Section 4 for the exact structure of the extracted cases. In the process of generating these cases we again used a small Java tool we developed to pre-process the raw retrieved LOD. During the use of this tool we discovered that it is very easy to further adapt this Tool to generate any desired kind of csv-formated files representing cases and thus upon later importing these files via myCBR, easily generate any kind of structural cases from LOD, extracted according to the desired structure of the cases. Similar to the possibility to rapidly built new vocabularies this approach of building structural cases from LOD is also a very rapid approach. Given that a basic tool for the exact extraction from the retrieved LOD is already in place the generation of new cases consists of just a new query to an LOD repository, a slight adjustment of the extraction tool and a new import of the generated csv-file via myCBR. This process can, if the cases are not overly complex, be performed in minutes and yield hundreds of cases only limited in their numbers by the amount of available LOD. Furthermore, using LOD also reduces the costs of knowledge acquisition in terms of time, logistics, effort, money, etc.

For the third knowledge container similarity measure we were able to build a taxonomy, carrying informations about its items similarity in the form of their distance within the taxonomy. The generated taxonomy contains 116 disease items an was built upon the input of 1000 category-disease-pairs. Top level structures in the generated taxonomy show a satisfying quality nevertheless in deeper levels of the taxonomy the quality of disease item links e.g. making 'sense' as father-child pairs of nodes deteriorates quickly. Also the amount of disease-items in the taxonomy is not satisfying. We assume this to be a result of a workaround we employed to generate the taxonomy. We used our recently developed tool KEWo which is specialized in processing natural language and information extraction out of unstructured text, generally discussion posts from an on-line forum. Feeding this tool with the highly structured text we derived from the LOD retrieval surely caused some conflicts with the original goal of the KEWo to extract information from unstructured text. We took this into account as far as we were aware of this 'missuse' and accepted the poorer results but thus getting at least a proof of concept taxonomy that shows that it is indeed possible to derive knowledge for the similarity measures from LOD. Also the process of generating a taxonomy from LOD is a bit more complex than the two other processes described in this paper it still is by far faster than a manual approach of building a taxonomy could not ever be. Despite the lack of quality regarding the linking of items in the deeper levels of the taxonomy, as a first proof of concept run, the generation of a similarity measure in form of a taxonomy can be seen as equally accelerated as the generation of the vocabulary and the cases are by the use of LOD.

6 Discussion

During our work with LOD we found it somewhat hard to identify relevant LOD repositories and the specific names of the attributes respectively predicates of the items in these repositories. Also we noticed there are ongoing efforts to improve the searchability of LOD this still is an issue we found somewhat hampering the use of LOD despite its ease and resource fullness if a correct repository and all of its facets are identified.

Nevertheless the rich and fast growing sources of LOD surely legitimate further research into their use as a source for knowledge to be used in the knowledge-containers of CBR. This is especially true if one takes into account their highly structured nature and the fact that LOD is available for free.

As far as we can comment on our first experimental results the use of LOD, once a working development environment consisting of querying tools, tools for further information refinement out of raw LOD and tools for incorporating this refined data into CBR knowledge containers like myCBR is present, filling the knowledge containers vocabulary, structured cases and similarity measure is an easy and fast process.

The fourth knowledge container of CBR that is containing adaptation knowledge, was not part of our experiments

and it is still questionable if there is a method to generate and/or extract knowledge for this container from LOD. We assume that there might be a chance to use the taxonomy we generated as a similarity measure to derive some adaptation knowledge from it, like the model-based adaptation approach presented in [Hanft *et al.*, 2010]. Furthermore, one can imagine that the structuring of much of the LOD as ontologies might yield some further sources for the extraction of adaptation knowledge from these ontologies.

7 Summary and Outlook

In this paper we proposed the idea to use LOD as a source to fill three out of the four knowledge containers a CBR system is based on. The containers we addressed were vocabulary, strucutred cases and the similarity measure. We have described our experimental setup regarding the methods used to acquire LOD in the field of diseases, the process of further refining the acquired data involving the development of some simple java-based tools and the use of more complex tools like our own developed KEWo and the external tools myCBR and Protégé to transform the LOD into fitting the structural requirements of the aforementioned knowledge containers [Bergmann, 2002].

We were able to produce good quantity and quality results for the knowledge containers vocabulary and structural cases as well as the data extracted for assigning similarity measures in form of a taxonomy, build upon the retrieved LOD, using our own tool KEWo is also acceptable. Further on, we were able to proof the fact that the approach of a semi-automatic acquisition of knowledge for the three mentioned knowledge containers from LOD resulted into a very rapid built up of these containers compared to manual or even other existing semi-automatic approaches we recently developed (see [Bach *et al.*, 2010b] for details).

Future goals based upon the work presented in this paper consist of the further refinement and automatization of the extraction process from LOD and the development of more efficient and even more easily customizable Tools for the Refinement of retrieved raw LOD. Also the question of how to extract adaptation knowledge should also be resolved to profit on the benefits of rapid knowledge acquisition for all four of the knowledge containers.

A distant future goal is given by the idea of incorporating all process-steps and tools presented in this paper into a single tool that handles the whole process from finding appropriate LOD resources, querying them, refining the retrieved data and formating it to be further processed by our own tools, like the KEWo, or external tools like myCBR. All of these functionalities should be embedded in a convenient GUI which allows the user to specify the knowledge container for which LOD should be retrieved for and the desired characteristics of this LOD, being most customizable in regard to the many facets each knowledge-containers contents can have.

References

[Althoff *et al.*, 2007] Klaus-Dieter Althoff, Kerstin Bach, Jan-Oliver Deutsch, Alexandre Hanft, Jens Mänz, Thomas Müller, Regis Newo, Meike Reichle, Martin Schaaf, and Karl-Heinz Weis. Collaborative Multi-Expert-Systems – Realizing Knowlegde-Product-Lines with Case Factories and Distributed Learning Systems. In Joachim Baumeister and Dietmar Seipel, editors, *Workshop Proceedings on the 3rd Workshop on Knowledge Engineering and Software Engineering (KESE 2007)*, Osnabrück, September 2007.

[Bach *et al.*, 2010a] Kerstin Bach, Meike Reichle, and Klaus-Dieter Althoff. Case-Based Reasoning in a Travel Medicine Application. In Isabelle Bichindaritz and Lakhmi Jain, editors, *Computational Intelligence in Medicine*, Advanced Information and Knowledge Processing, page to appear. Springer, 2010.

[Bach *et al.*, 2010b] Kerstin Bach, Christian Severin Sauer, and Klaus-Dieter Althoff. Deriving Case Base Vocabulary from Web Community Data. In Cindy Marling, editor, *ICCBR-2010 Workshop Proceedings: Workshop on Reasonng From Experiences On The Web*, page to appear, 2010.

[Bergmann, 2002] Ralph Bergmann. *Experience Management: Foundations, Development Methodology, and Internet-Based Applications*, volume 2432 of *Lecture Notes in Computer Science*. Springer, 2002.

[Bizer *et al.*, 2008] Christian Bizer, Tom Heath, Kingsley Idehen, and Tim Berners-Lee. Linked data on the web (LDOW2008). In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 1265–1266, New York, NY, USA, 2008. ACM.

[Hanft *et al.*, 2010] Alexandre Hanft, Régis Newo, Kerstin Bach, Norman Ihle, and Klaus-Dieter Althoff. CookIIS - A successful Recipe Advisor and Menu Advisor. In Stefania Montani and Lakhmi Jain, editors, *Successful Case-based Reasoning applications*. Springer, 2010.

[Plaza, 2008] Enric Plaza. Semantics and Experience in the Future Web. In Klaus-Dieter Althoff, Ralph Bergmann, Mirjam Minor, and Alexandre Hanft, editors, *Advances in Case-Based Reasoning, 9th European Conference, ECCBR 2008, Trier, Germany, September 1-4, 2008. Proceedings*, volume 5239 of *Lecture Notes in Computer Science*, pages 44–58. Springer, 2008.

[Reichle *et al.*, 2009] Meike Reichle, Kerstin Bach, and Klaus-Dieter Althoff. The SEASALT Architecture and its Realization within the docQuery Project. In Bärbel Mertsching, editor, *Proceedings of the 32nd Annual Conference on Artificial Intelligence (KI-2009)*, Lecture Notes in Informatics, pages 556–563, September 2009.

[Richter, 1998] Michael M. Richter. Introduction. In Mario Lenz, Brigitte Bartsch-Spörl, Hans-Dieter Burkhard, and Stefan Wess, editors, *Case-Based Reasoning Technology – From Foundations to Applications*, LNAI 1400. Springer-Verlag, Berlin, 1998.

[Sauer, 2010] Christian Severin Sauer. Analyse von Webcommunities und Extraktion von Wissen aus Communitydaten für Case-Based Reasoning Systeme. Master's thesis, Institute of Computer Science, University of Hildesheim, 2010.

[Stahl and Roth-Berghofer, 2008] Armin Stahl and Thomas R. Roth-Berghofer. Rapid Prototyping of CBR Applications with the Open Source Tool myCBR. In *ECCBR'08: Proceedings of the 9th European conference on Advances in Case-Based Reasoning*, pages 615–629, Berlin, Heidelberg, 2008. Springer-Verlag.

Memex360 – Persönliches Wissensmanagement mit Theseus ORDO

Björn Decker
Ralph Traphöner
Attensity Europe GmbH
Europaallee 10, 67657 Kaiserslautern, Germany
bjoern.decker@attensity.com ralph.traphoener@attensity.com

Abstract

Die Herausforderung, sich in der beständig zunehmenden Information zurechtzufinden, ist nicht neu. Durch verschiedene Applikationen - insbesondere des Web 2.0 - sind Lösungen entstanden, die einen Benutzer dabei unterstützen, einen Teil der für ihn relevanten Information zu verwalten und somit Wissen zu generieren. Allerdings bleiben die jeweiligen Informationen auf die jeweilige Applikation beschränkt. Mit Memex360 - einem Ergebnis des Use Cases ORDO im Forschungsprogramm THESEUS - wird eine integrative Schicht auf die Informationen implementiert, wobei die jeweiligen Werkzeuge weiter genutzt werden können. In dem Beitrag wird die Idee und Architektur von Memex360 vorgestellt.

1. Einleitung

Die Herausforderung, sich in der beständig zunehmenden Menge an Information zurechtzufinden, ist nicht neu. Bereits 1945 veröffentlichte Vanevar Bush die Idee des MEMEX [Bush, 1945]. Dieser MEMEX (memory extender) unterstützt einen Wissensarbeiter dabei, die für ihn relevante Information zu erfassen, zu annotieren und miteinander in Bezug zu setzen. Grundidee dabei war es, den Menschen bei der Verwaltung und Erschließung seines Wissens zu unterstützen – nicht den Menschen zu ersetzen.

Eine Vielzahl von Web 2.0 Applikationen ermöglicht es heute, Informationen zu erfassen, zu annotieren und mit einander in Bezug zu setzen stehen. Blogs erlauben es, persönliche Notizen zu erfassen. Soziale Netzwerke helfen dabei, passende Ansprechpartner zu finden. Und nicht zuletzt Suchmaschinen – sei es auf dem Desktop oder im Web – sind in der Lage, umfangreiche Datenbestände schnell zu erschließen.

Allerdings stehen diese Lösungen in der Regel für sich allein. Erschließt sich ein Nutzer ein neues Wissensgebiet, kann er zum Beispiel eine Notiz zu einem Dokument in einem Blogbeitrag anlegen, oder in einer Annotation in dem jeweiligen Dokument oder der Notiz-Funktion des Bookmark-Dienstes. Welche Form der Annotation genutzt wird, hängt von den Vorlieben des Benutzers und seiner Situation ab – die Notiz bei seinem bevorzugtem Bookmark-Dienst wird z.B. unterwegs benutzt, während die Notiz in einem Dokument genutzt wird, wenn er keine Internet-Verbindung hat. Diese Annotationen stehen also nur in der jeweiligen Applikation zur Verfügung – die jeweilige Information bleibt also isoliert.

Doch nicht nur die Informationen in den Dokumenten bleiben isoliert. Auch das Wissensmodell, mit denen sich ein Benutzer Dokumente erschließt und klassifiziert, liegen in der Regel nur in der jeweiligen Applikation vor. Die in einem Personal Information-Management (PIM)-Werkzeug wie Outlook angelegten Kategorien liegen nur hier vor – um damit Blog-Einträge oder Bookmarks zu kategorisieren, müssten die Kategorien in der jeweiligen Applikation manuell gepflegt werden. Aufgrund des damit verbundenen hohen Aufwandes unterbleibt dabei meistens ein Abgleich der Kategorien.

Zusätzlich hat ein Nutzer in den verschiedenen Applikationen Informationen, welche die Erschließung von Dokumenten weiter unterstützen können. Beispielsweise können die Kontaktdaten im PIM oder in sozialen Netzwerken genutzt werden, um Firmennamen und Personen in relevanten Dokumenten zu identifizieren.

Die technische Grundlage, um diese Informationen mittels einer integrativen Schicht miteinander in Bezug zu setzen, liegt in den technischen Standards, mit denen die Inhalte in den jeweiligen Applikationen angesprochen werden können. Im Bereich der Blogs haben sich zum Beispiel die Blogger API [Winer, 2002] und die Meta-Weblog API [Anonymous] etabliert, um sowohl Inhalte und Kategorien abzufragen als auch zu erstellen. Durch RSS [Winer, 2003] Benachrichtigungen existiert ein Mechanismus, um Änderungen in einer Applikation an andere Applikationen oder den Benutzer zu kommunizieren. So können aktuelle Änderungen zeitnah von Suchdiensten in einer maschinell verarbeitbaren Form erfasst werden.

Memex360 implementiert diese integrative Schicht, welche die von einem Benutzer verwendeten Quellen integriert, miteinander in Bezug setzt und die Ergebnisse an den Benutzer meldet.

2. Memex360 – Idee und Architektur

Der Ausgangspunkt von Memex360 ist, dass der Benutzer seine bisher verwendeten Werkzeuge zum persönlichen Wissensmanagement weiterhin verwenden kann. Memex360 sorgt für die entsprechende Integration der unterschiedlichen Applikationen. Die entsprechende Architektur ist in der folgenden Abbildung dargestellt. Die obere Schicht (Mashups und Browser Integration) deckt mögliche Werkzeuge zum persönlichen Wissensmanagement ab. Von links nach rechts sind dies:

- PIM-Werkzeuge und RSS-Feed-Aggregatoren, um dem Benutzer beim Selbstmanagement zu unterstützen und über Neuigkeiten auf dem laufenden zu halten.

- Bookmarks-Verwaltungs-Werkzeuge, um Referenzen auf relevante Dokumente zu erfassen.
- Blogging oder Wikis Tools, um persönliche Notizen zu erfassen
- Visualisierungs-Tools, z. B. Mindmaps, um Informationen miteinander in Bezug zu setzen.

Mittels standardisierter Zugriffs-Mechanismen werden zum Einen die Inhalte dieser Applikationen durch die Service-Plattform erfasst und aufbereitet. Zum Anderen werden die Inhalte – wie ähnliche Dokumente oder Klassifikationen - in der Service-Plattform den verschiedenen Applikationen zur Verfügung gestellt. Die Service-Plattform hat dabei die folgenden vier grundlegenden Dienste:

- Der *Such-Service* indiziert zum Einen die relevanten Dokumente und reichert sie mit Metadaten an. Zum Anderen stellt dieser Service eine föderierte Suche zur Verfügung, um weitere Informationsquellen – durch Wolken angedeutet - zu integrieren. Neu hinzugekommene Informationen werden mittels konfigurierbarer RSS Feeds an den Nutzer weitergeleitet.
- Der *Referenz-Dienst* verwaltet die Referenzen des Benutzers auf relevante Dokumente. Weiterhin leitet der Referenz-Dienst neue Referenzen an den Such-Service zur weiteren Bearbeitung weiter.
- Der *Zettelkasten-Dienst* erfasst persönliche Notizen und setzt sie mit den Dokumenten aus dem Referenz-Dienst in Bezug.
- Den *Ontologie-Dienst*, um das persönliche Wissensmodell des Benutzers zu pflegen und den anderen Diensten zur Verfügung zu stellen. Die Ontologie wird entweder durch Hinzunahme von Metadaten in den Applikationen (z.B. neue Kategorien) oder dem Visualisierungs-Werkzeug gepflegt. Vorschläge für Erweiterungen der Ontologie können dabei durch die Analyse des Such-Services erstellt werden.

3. Die Erstellung einer Marktanalyse – Ein Beispiel für die Nutzung von Memex360

Herr X ist mit der Erstellung einer Marktanalyse zum Thema „Capsaicin“ beauftragt – einem Bestandteil des Chillis, welcher z.B. in der Schmerztherapie genutzt wird. Dazu nutzt er die Mindmap, welche er zu einem ähnlichen Thema bereits angelegt hatte. Ausgangspunkt sind die Fragen der bereits durchgeführten Marktanalyse zu Definitionen, ähnlichen Wirkstoffen, Studien und Anbieter. Die entsprechenden bereits angelegten Kategorien und Klassen in der Ontologie kann er wiederverwenden. Die neuen Einträge werden darüber hinaus mit der aktuellen Aufgabenstellung als Arbeitskontext verschlagwortet. Herr X sucht nun nach entsprechenden Begriffen – und richtet gleichzeitig entsprechende Benachrichtigungen auf dem Such-Service ein. So wird er später über Änderungen informiert. Relevante Treffer merkt er sich im Referenzdienst und fertig eigene Notizen an. Via RSS stehen diese Notizen und Referenzen auch im Mindmap Tool zur Verfügung und können hier mit anderen Ergebnissen seiner

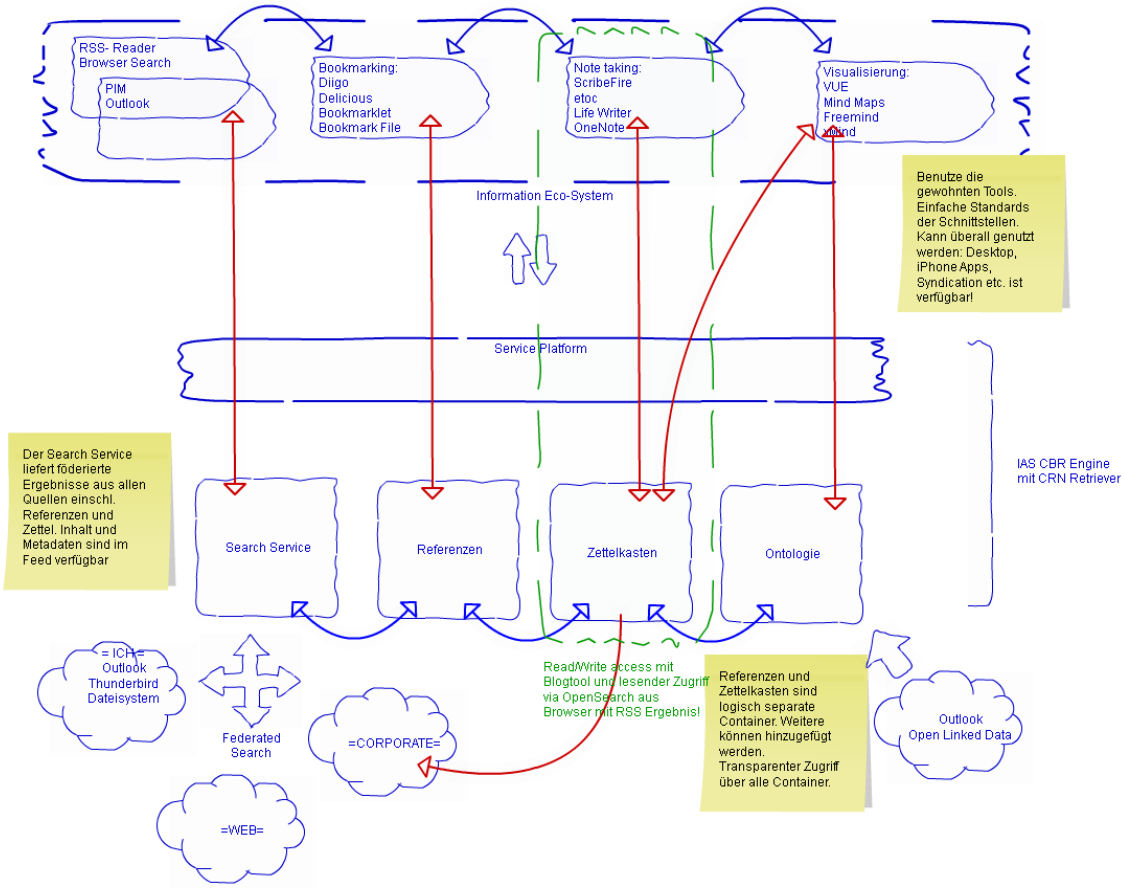


Abbildung 1: Memex360 Architekturskizze

Recherche in Bezug gesetzt werden.

Bei der Marktübersicht erstellt er unter anderem einen Überblick über Forschungsinstitute und Herstellerfirmen – jeweils von Herrn X angelegten Unterklassen von Anbieter. Vorschläge hierzu werden aufgrund der linguistischen Analyse der gefundenen Objekte – oder bereits vorhandenen Metadaten – unterbreitet.

Bestandteil der o.g. Klassen ist dabei eine Referenz auf eine Instanz der Klasse Ort. Auch hierfür können für den Anbieter entsprechende Vorschläge generiert werden. Die Ortsangabe wiederum kann genutzt werden, um über Open Linked Data (z. B. geonames.org) weitere Informationen zu ermitteln. So können die Geokoordinaten hinzugefügt werden, um die räumliche Verteilung der Anbieter zu analysieren.

Während der Recherche verfeinert Herr X auch an anderen Stellen sein Wissensmodell. Er findet z. B. immer wiederkehrende Personen, welche er als wichtige Autoren erkennt. Diese kann er z.B. in seinem Adressbuch ablegen und so in auch in die Ontologie einfügen. Auch richtet er neue Unterkategorien ein. Diese Erweiterungen können dann genutzt werden, um mit den bestehenden Anfragen weitere relevante Dokumente zu suchen und zu identifizieren.

Durch die zunehmende Anreicherung der Mindmap können darüber hinaus weitere, ähnliche Einträge in anderen Mindmaps identifiziert und zur Integration in die aktuelle Mindmap vorgeschlagen werden.

4. Prototypische Implementierung

Die Implementierung des Memex360 Prototyps hat den Status eines Proof of Concept. Dabei wird der Suchdienst durch eine CBR[1] Komponente zur Verfügung gestellt, d. h. es handelt sich im Kern um eine strukturierte fallbasierte Anwendung.

Alle Informationsobjekte, seien es Webseiten oder eigene Notizen und Dokumente, werden als strukturierte Fälle repräsentiert. Die erforderlichen Metadaten werden durch die Verschlagwortung des Anwenders und Ontologie basierte Informationsextraktion aus dem Inhalt des Informationsobjekts gewonnen. Letzteres erfolgt mit einer auf linguistischen Regeln basierenden Text Mining Engine (TME), die auch im Rahmen von THESEUS ORDO entwickelt wird.

Das Schema – die Ontologie – der strukturierten CBR Engine kann domänenspezifisch modelliert werden. Dabei ist es weitgehend unerheblich welcher Formalismus verwendet wird. Weitgehend in dem Sinne, das die genutzte IAS objektorientierte Modelle nutzt, die eine extensionale Klassendefinition voraussetzen. Eine bijektive Abbildung in OWL ist nicht möglich, da nicht alle OWL Konstrukte unterstützt werden. SKOS und RDFS Modelle sind abbildbar und Modelle können importiert werden.

Modellierung durch den Anwender bezieht sich dabei immer auf Instanzen. Schema, d. h. Klassen, können nicht zur Laufzeit bearbeitet werden.

Bewusst wurde Memex360 so ausgelegt, dass die Benutzung nicht an dedizierte Clients gebunden ist. Vielmehr bettet sich Memex360 nahezu unsichtbar in die bestehende Arbeitsumgebung ein:

- Referenzen und Bookmarks können mit allen gängigen Tools verwaltet werden, soweit diese ihre Inhalte als RSS Feeds zur Verfügung stellen. Die Autoren nutzen Delicious[2] und Zotero[3].
- Neue Bookmarks werden als solche erkannt und automatisch durch Crawler oder RSS Abonnement zur Fallbasis hinzugefügt.
- Die direkte Erstellung eigener Inhalte und Notizen kann mit Blogwerkzeugen erfolgen. Hierzu wurde Memex360 um die gängigen Blog API Protokolle erweitert. Damit wird z. B. Windows Live Writer zu einem komfortablen Editor für die eigene Fallbasis.
- Die Verschlagwortung von Inhalten erfolgt jeweils durch die nativen Tagging-Mechanismen der vorgenannten Werkzeuge. Wird dabei in z.B. Delicious ein neues Tag vergeben, so wird es in die Ontologie aufgenommen, steht dann im Live Writer als Schlagwort zur Verfügung und kann auch dort in den Kategoriebaum eingeordnet werden, der wiederum von der CBR-Engine als Ähnlichkeitsmaß interpretiert wird.
- Die Suche nach Inhalten erfolgt mit Hilfe des Open Search Interface direkt aus der Suchbox des Browsers. Dort wird Memex360 genauso als Suchmaschine zur Verfügung gestellt wie Bing oder Google. Die Treffer zu einer Suche werden als RSS Feed[4] zurückgeliefert, der auch die strukturierten Falldaten enthält. Dies ermöglicht es insbesondere auch, eine Suche als Abonnement zu einem Standard RSS Feed Reader hinzuzufügen und damit bei neuen Treffern zur gespeicherten Suche auf dem Laufenden zu bleiben.
- Für die im Marktanalyse Szenario genannte Mindmap Funktionalität wurde die Visual Understanding Environment (VUE)[5] genutzt. VUE verarbeitet RSS Feeds und erlaubt das strukturieren und verbinden von Inhalten und Metadaten per Drag und Drop aus einem RSS Objekt. Beinhaltet ein Fall im Trefferfeed z. B. eine neue erkannte Person, so kann diese als Konzept auf die Arbeitsfläche gelegt und anschließend mit der Klasse Personen verbunden werden. Noch zu realisieren ist ein Rückkanal von VUE in die Ontologie des Memex360, um modellrelevante Änderungen automatisch zu übernehmen. Alternativ wird zurzeit der Einsatz von Wise Mapping[6] als Web basiertes Modellierungs- und grafisches Recherchewerkzeug geprüft.

Eine weitere wünschenswerte Erweiterung ist das Abonnement von Quellen für Instanzen aus der sogenannten Open Linked Data Cloud. Aktuell erfordert dies noch einen expliziten Import aus RDF Dateien oder einem Triplestore.

[1] Case-Based Reasoning

[2] http://www.delicious.com

[3] http://www.zotero.org

[4] RSS als Protokoll stellt kein Sicherheitsrisiko dar, solange das Memex360 Backend im Unternehmensnetzwerk betrieben und per VPN zugegriffen wird

[5] http://vue.tufts.edu/

[6] http://www.wisemapping.org/

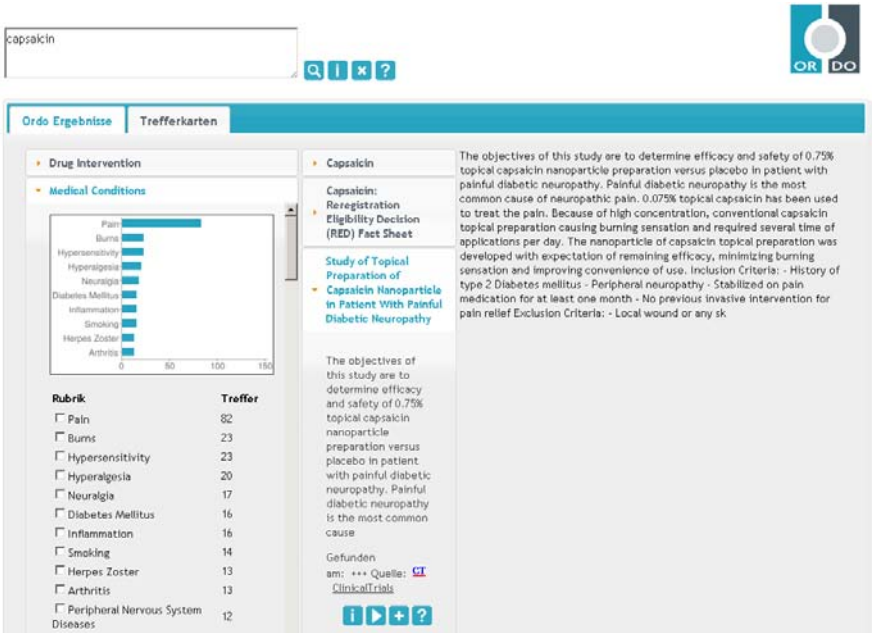


Abbildung 2: Memex360 - Suche nach Capsaicin mit JavaScript Interface

Neben der Benutzung über Werkzeuge mit Schnittstellen mit Web 2.0 Standards verfügt Memex360 über eine REST API im JSON Datenformat. Auf diese Schnittstelle können – in Ergänzung zur Benutzung durch die Standardwerkzeuge dedizierte Benutzerschnittstellen aufgesetzt werden. Mit dem JSON/REST API wurde z. B. eine dedizierte Suchschnittstelle für einen Demonstrator in der Life Science Domäne implementiert. Abbildung 2 zeigt das Interface, welches mit JQuery realisiert wurde.

5. Vergleichbare Ansätze

Mit diesen Funktionalitäten hat Memex360 das persönliche Wissensmanagement im Fokus. Aus Platzgründen werden im Folgenden zwei ausgewählte ähnlich gelagerte Ansätze präsentiert, welche aus Sicht der Autoren die stärkste Ähnlichkeit zum Memex360 Konzept aufweisen: Semantische Wikis und Semantische Desktops.

Durch semantische Wikis (z.B. [Oren et al 2006]) können persönliche Notizen als Wiki-Seiten – und damit mit einem eigenen GUI - mit Metadaten angereichert und durch Links miteinander in Bezug gesetzt werden. Externe Informationsobjekte können verlinkt werden. Damit kann ein semantischer Wiki Funktionen der zuvor genannten Standard-Applikationen übernehmen (z.B. Bookmarks in einer Liste strukturieren). Semantische Desktops (z.B. [Sauermann, 2005]) stellen wiederum eine Infrastruktur und entsprechende GUI-Komponenten zur Verfügung, um Informationsobjekte mit semantischen Annotationen (Metadaten und Relationen) miteinander in Bezug zu setzen. Im Fokus dieser Auszeichnung sind dabei die auf dem Desktop verfügbaren Informationen. Beide Applikationsklassen enthalten dabei sowohl relevante Daten als auch Informationen über das Wissensmodell des Benutzers. Aus Sicht von Memex360 können diese Applikationsklassen als weitere Werkzeuge fungieren, um den Wissensraum des Benutzers zu erschließen. Sie stellen damit weitere Komponenten im Informationsökosystem dar, die aber nicht Memex360 als Monitor des Systems ersetzen.

6. Zusammenfassung und Ausblick

Memex360 integriert die unterschiedlichsten Informationsquellen und baut dabei auch Brücken zwischen diesen Quellen. Damit wird durch Memex360 der Dispersion der persönlichen Information begegnet.

Die vorgestellten Ideen und Konzepte werden im Rahmen des THESEUS ORDO Projektes weiterentwickelt.

Danksagung

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Wirtschaft und Technologie unter dem Förderkennzeichen 01MQ07005 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt bei den Autoren.

Literatur

[Anonymous] Anonymous. Blogger APIs - Google Code. http://code.google.com/apis/blogger/.

[Bush, 1945] Bush, Vanevar. 1945. As We May Think - Magazine - The Atlantic. The Atlantic (Juli): 47-61. http://www.theatlantic.com/magazine/archive/1969/12/as-we-may-think/3881/.

[LOD] Linked Data Org. Linked Data | Linked Data - Connect Distributed Data across the Web. http://linkeddata.org/.

[Winer, 2002] Winer, Dave. 2002. RFC: MetaWeblog API. März 14. http://www.xmlrpc.com/metaWeblogApi.

[Winer, 2003] Winer, Dave. 2003. RSS 2.0 Specification (RSS 2.0 at Harvard Law). Juli 15. http://cyber.law.harvard.edu/rss/rss.html.

[Oren et al 2006] Oren, Eyal et al. 2006. Semantic Wikis for Personal Knowledge Management (Database and Expert Systems Applications, LNCS). Springer Berlin

[Sauermann, 2005] Sauermann, Leo. 2005. The semantic desktop-a basis for personal knowledge management (Proceedings of the I-KNOW).

ABIS 2010 - 18th Intl. Workshop on Personalization and Recommendation on the Web and Beyond

Melanie Hartmann
Technische Universität Darmstadt
Darmstadt, Germany
melanie@tk.informatik.tu-darmstadt.de

Eelco Herder Daniel Krause
L3S Research Center
Hannover, Germany
{herder, krause}@L3S.de

Andreas Nauerz
IBM Research and Development
Böblingen, Germany
andreas.nauerz@de.ibm.com

1 The ABIS Workshop Series

ABIS 2010 is an international workshop, organized by the German SIG on Adaptivity and User Modeling[1] of the Gesellschaft fr Informatik. ABIS takes place in connection with LWA2010, together with related workshops on information retrieval, machine learning and information management.

2 About ABIS 2010

Personalization has become a core feature on the Web and beyond: Google provides personalized search results. Amazon recommends books and other products. Facebook suggests friends and groups. Personalized features and recommendations include items that were appreciated by similar users or the users friends and are typically based on a users profile data, the users current location or items that the user browsed, searched, tagged or bought earlier. Mashups and cross-application linking of user profiles promise to provide even more relevant suggestions and services.

Personalization is great. But personalization can go awfully wrong, too. Systems may draw wrong conclusions about your search actions and constantly annoy you with personalized menus that do not work or recommendations for books that you couldnt care less for. And do you really want your friends and colleagues to know what products you searched for yesterday?

Our keynote speaker will be Jürgen Geck, CEO of OpenXchange[2], a company that provides integrated tools for mail and collaboration to millions of users in Europe.

3 Workshop Overview

The program commitee received submissions from research and industry within the broad area of User Modeling and Adaptive Systems. Special emphasis of this year's workshop was on submissions in the following areas:

- *Obtaining user data*: logging tools, aggregation of data from social networks and other Web 2.0 services, location tracking
- *Modeling the user data*: collaborative filtering, cross-application issues, contextualization and disambiguation, use of ontologies and folksonomies
- *Personalization and recommendation*: applications in social networks, search, online stores, mobile computing, e-learning and mash-ups
- *Evaluation and user studies*: laboratory studies, empirical studies and analysis of existing corpora of usage data
- *Emerging and important issues*: future applications, new paradigms in human-computer interactions, privacy awareness

4 Program Chairs

- Melanie Hartmann (Technische Universität Darmstadt, Germany)
- Eelco Herder (L3S Research Center, Germany)
- Daniel Krause (L3S Research Center, Germany)
- Andreas Nauerz (IBM Research and Development, Germany)

5 Program Committee

- Fabian Abel (Leibniz University Hannover, Germany)
- Maria Bielikova (Slovak University of Technology, Slovakia)
- Susan Bull (University of Birmingham, UK)
- Betsy van Dijk (University of Twente, Netherlands)
- Birgitta Knig-Ries (Universität Jena, Germany)
- Peter Dolog (Aalborg University, Denmark)
- Eelco Herder (L3S Research Center, Germany)
- Sabine Graf (Athabasca University, Canada)
- Melanie Hartmann (Technische Universität Darmstadt, Germany)
- Dominikus Heckmann, DFKI Saarbrücken, Germany)
- Nicola Henze (Leibniz University Hannover, Germany)
- Vera Hollink (Centrum Wiskunde & Informatica, The Netherlands)
- Daniel Krause (L3S Research Center, Germany)
- Tsvi Kuflik (University of Haifa, Israel)
- Erwin Leonardi (Universiti Tunku Abdul Rahman, Malaysia)
- Andreas Nauerz (IBM Research and Development, Germany)
- Alexandros Paramythis (Johannes-Kepler-University, Linz, Austria)
- Wolfgang Reinhardt (Universität Paderborn, Germany)
- Sven Schwarz (DFKI, Germany)
- Marcus Specht (Open University of the Netherlands, Netherlands)
- Stephan Weibelzahl (National College of Ireland)

[1] http://abis.l3s.de
[2] http://www.open-xchange.com/

What is wrong with the IMS Learning Design specification? Constraints And Recommendations

Daniel Burgos[1, 2]
[1]Atos Origin. Research & Innovation
Albarracin 25, Madrid 28027, Spain
daniel.burgos@ atosresearch.eu
www.atosresearch.eu
[2]International University of La Rioja
Gran Via Rey Juan Carlos I 41, Logroño, La Rioja 26002, Spain
www.unir.net

Abstract

The work presented in this paper summarizes the research performed in order to implement a set of Units of Learning (UoLs) focused on adaptive learning processes, using the specification IMS Learning Design (IMS-LD). Through the implementation and analysis of four learning scenarios, and one additional application case, we identify a number of constraints on the use of IMS-LD to support adaptive learning. Indeed, our work in this paper shows how IMS-LD expresses adaptation. In addition, our research presents a number of elements and features that should be improved and-or modified to achieve a better support of adaptation for learning processes. Furthermore, we point out to interoperability and authoring issues too. Finally, we use the work carried out to suggest extensions and modifications of IMS-LD with the final aim of better supporting the implementation of adaptive learning processes.

1 A brief description of the IMS Learning Design

IMS Learning Design (or simply IMS-LD) [IMS, 2003] is aimed to transform regular lesson plans into interoperable Units of Learning (UoL). This specification is able to use any pedagogical model to get a UoL run-able and editable in an interoperable way. IMS-LD augments other well-known e-learning specifications aforementioned, like SCORM, IMS Content Packaging, IMS Question and Test Interoperability or IMS Simple Sequencing. Furthermore, IMS-LD provides a language to describe the teaching and learning process in a Unit of Learning. It describes among other things the roles, the activities, the basic information structure, the communication among different roles and users; and all these under the pedagogical approach decided by the teacher and-or the learning designer. In this section, we show what is IMS-LD and how it is structured, as well as how it provides Adaptation within the UoLs

IMS-LD is able to describe a full learning flow with several elements -such as roles, activities, environments or resources- and features -such as properties, conditions, monitoring services or notifications [Burgos & Griffiths, 2005; Koper & Tattersall, 2005].

The usual life-cycle starts with a lesson plan modelled according to the IMS-LD specification, defining roles, learning activities, services and several other elements, inside an XML document called Manifest. An information package written in IMS Content Packaging [IMSCP, 2001] is used as a container for the resources and links them with the IMS-LD structure. Later, the Manifest is packaged with the nested resources in a compressed ZIP file, meaning a UoL. Several examples available are shown later on.

IMS Learning Design uses the metaphor of a theatrical play to visualize how to model Units of Learning. A play is performed by a number of actors, who may take up a number of roles at different times in the play. Similarly in learning design a learner can take up different roles at different stages of a learning process. At the end of each act the action stops, all the learners are synchronised, and then a something new can begin.

IMS-LD consists of three levels: Level A, with the definition of the method, plays, acts, roles, role-parts, learning activities, support activities and environments. It is the core of the specification, contains the description of the elements that configure IMS LD and the coordination between them. For instance, role-parts define what activities must be taken by a role in order to complete an act and, subsequently, a play.

Level B, adds properties, conditions, calculations, monitoring services and global elements to Level A, and provides specific means to create more complex structures and learning experiences. Properties can be used as variables, local or global ones, storing and retrieving information for a single user, a group or even for all the characters involved. Through these mechanisms the learning flow can be changed at the run time, as decisions can be made taking into account dynamic content. Logically it is the used level to express the most of the pedagogical needs concerning Adaptation, personalization, feedback, tracking and several other usual requests of teachers and learning designers.

Finally, Level C adds notifications to Level B, meaning an email sent and a show/hide command to a specific activity, depending on the completion of another one [Koper & Burgos, 2005].

2 IMS-LD and Adaptation

In addition to the basic structure of Level A, the elements in Level B and Level C are actually the key for more expressive UoLs (for instance, based on Adaptation or Collaboration), as they combine several features that encourage and make the content and the learning flow more flexible [Koper & Burgos, 2005]. Furthermore, the combination of these elements allows for the modelling of several classical adaptive methods (i.e. reuse of pedagogical patterns, adaptability, navigational guidance, collaborative learning, contextualized and mobile distributed learning, Adaptation to stereotypes), making use of different structural elements of IMS-LD, like i.e. Environment, Content, User groups and Learning flow [Burgos et al., 2007].

In a literature study, we identify eight different kinds of Adaptation being carried out in eLearning systems [Burgos, 2008]: Interface based, Learning flow based, Content based, Interactive problem solving support, Adaptive information filtering, Adaptive user grouping, Adaptive evaluation, and Changes on-the-fly. All of them use various inputs provided during the learning process and aim to tune the activities and actions of the learner to get the best learning experience as possible [Butz et al., 2003]. A wide and consistent set of rules of dependencies among users, methods and learning objects is needed to describe these eight types of Adaptation, and moreover their possible combinations. If we categorize all these types of Adaptation, we can group them in two clusters [Ahmad et al., 2004; Chin, 2001; De Bra et al., 2004; Baeza-Yates & Ribeiro-Nieto, 1999; Van Rosmalen & Boticario, 2005; Merceron & Yacef, 2003; Romero et al., 2003]. The first one consists of three types of Adaptation:

1. Interface-based (also called adaptive navigation and related to usability and adaptability) where elements and options of the interface are positioned on the screen and their properties are defined (color, size, shadow, etc.); this is closely related to general customization and supporting people with special needs which influence personalization, such as colour impairment or poor hearing, for instance.
2. Learning flow-based, where the learning process is dynamically adapted to sequence the contents of the course in different ways. The learning path is dynamic and personalised for every student, but even also for every time that the course is started (also called run or instance), so that the student can take a different itinerary depending on his performance.
3. Content-based, where resources and activities dynamically change their actual current content, as in Adaptive and Intelligent Web-Based Educational Systems based on adaptive presentation [Brusilovsky & Miller, 2001]. For instance, the information inside a learning activity can be classified in three levels of depth, and every level is shown based on a number of factors.

The first cluster with three types of Adaptation becomes the base for the next one. Additional kinds of Adaptation feed a second cluster: 4) Interactive problem solving support; 5) Adaptive information filtering, 6) Adaptive user grouping; 7) Adaptive evaluation; and 8) Changes on-the-fly.

3 Methodology of analysis

This section describes how we have carried out the analysis, as well as the methodology followed to do the research in this paper. Previously, we have described how adaptation is envisaged by IMS-LD and which types of adaptation can be expresses with this specification. Furthermore, we have described, modelled and implemented a number of Learning Scenarios which show features for adaptive learning processes.

First, we have defined, modelled and analysed five Units of Learning (UoLs), which are described as learning scenarios (Table 1). In these learning scenarios, we describe adaptive learning processes and features. Further, we carried out an analysis of a real application case from the ATOS University, where a Unit of Learning (UoL) with adaptation features modelled with IMS-LD, was implemented (Figure 1).

ID	Type of adaptation	Description
1	Adaptive Assessment	adaptation on the learner´s performance and knowledge
2	Adaptive Authoring	adaptation on the learning designer´s method
3	Adaptive Content	adaptation on the learner´s decision
4	Adaptive Mentoring	adaptation on the teacher´s decision
5	Combination of adaptive types	Application case on Corporate training

Table 1. Learning scenarios

Last, every learning scenario is analysed and reports on shortcomings and recommendations to improve the expressiveness of IMS Learning Design to achieve a better adaptation process.

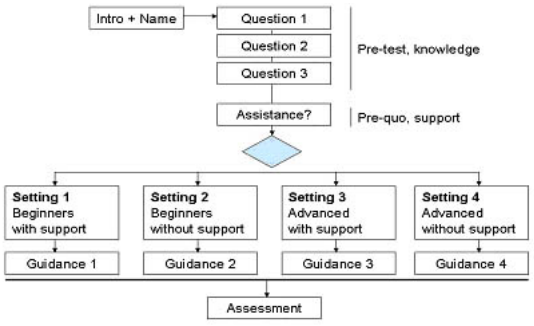


Figure 1. **ATOS application case**

Our analysis is focused on the main challenges and limitations to performing adaptive learning with IMS-LD. These mainly focus on the need for improving the flexibility and interoperability of this specification, while modelling adaptation.

All of them are available at the GRAPPLE Project website[1].

[1] www.grapple-project.org

4 How IMS-LD expresses Adaptation

In this section, we examine how IMS-LD can be used to represent each of the eight types of Adaptation aforementioned. A combination of the following proposals on Adaptation could support the performance of every role in an eLearning process. Taking the first block (which consists of the three main types), IMS-LD is able to model Adaptation:

4.1 Adaptation based on the interface

Interface Adaptation is based on options, navigation and visualization facilities. Interface Adaptation is not possible with today's tools for IMS-LD, such as CopperCore[2] Player, Reload LD Player and Sled. As long as the Adaptation of the interface is based on the tool and not on the Unit of Learning that is interpreted by the player, this is still true. Today's players do not yet provide facilities to change the size or the position of the navigation panels, or even open and close the working areas in the player. Either, these tools cannot change the style sheets related to a HTML file, part of the content, and any of the linked features, as font-size, font-type or background colour, for instance. Although the CopperCore engine provides the appropriate infrastructure, no player uses it so far. Nevertheless, some kind of adaptive interface is possible, using DIV layers and environments.

4.2 Adaptation based on the learning flow

The modification of the learning flow as the Unit of Learning is being executed is one of the most often used types of Adaptation. Taking the flow as a base, the Unit of Learning provides different activities, resources and services, depending on these four inputs during execution (user´s behavior and performance, user´s decision, teacher and set of rules). The activity structure in an IMS-LD UoL is defined using plays, acts, activity structures, learning activities, support activities and environments. We can also use the property of visibility to hide and show these elements and to adapt the learning flow. In these cases the property works as a flag, switching on and off the elements referred to.

4.3 Adaptation based on the content

Content Adaptation is based on the information inside an activity that is shown and handled. We know that a learning flow is mainly focused on the sequence of the activities in a Unit of Learning. However, content based Adaptation is focused on the information of every activity, and on the activity itself. There are two main approaches for content based Adaptation in IMS-LD: Flag properties and content of properties. Flag properties hide and show elements like e.g. activities or environments. On the other side, the content of specific properties can be modified on the run, making use of global elements in the specification.

4.4 Elements in Levels B and C to model Adaptation

The elements in Level B and Level C providing support to Adaptation in Units of Learning are categorized as a) properties, b) conditions, c) global elements, d) calculations, e) monitoring services, and f) notifications [Koper & Burgos, 2005; Burgos & Specht, 2006]:

1. Definition, set-up and use of properties: Properties are taken as variables to store values. There are several types of properties: local, local-personal, local-role, global-personal, global. There is also a property-group that is able to compile a number of the others.
2. Conditions: IMS-LD is able to define a basic structure if-then-else, or multiple structure with several chained basic if-then-else in a row, for instance to change the value of a property or to show and hide one element.
3. Global elements: Global elements provide a communication flow between the imsmanifest.xml, where the different levels of IMS-LD are set-up, and other XML files. Mainly, they can get an input from the user and they can show a value of a property. Furthermore, they can manage DIV layers in XHTML, for instance to show and hide specific content.
4. Calculations: IMS-LD is able to make some basic arithmetic's (sum, subtraction, multiplication and division) and some combination of a number of them in a row, to get a more complex formula, like a simple average, for instance.
5. Monitoring service: The specification allows monitoring any kind of property assigned to a user or a role, for instance. In order to start this action, firstly the component monitor must be set-up inside an environment and later the property can also be monitored.
6. Notifications: An action is automatically launched depending on the state of a property or a previous action, i.e., when a student ends an assignment an email is sent to the tutor.

5 Identification of constraints, gaps and issues to cope with

We use every learning scenario aforementioned as a base to find restrictions, drawbacks and elements to improve within the specification. These resources show how far IMS-LD supports adaptation, when different inputs and roles are involved. We also make links to the integration of UoLs, when needed. Out of the modelling and development of those UoLs we perform an analysis on which features, elements and components are missing or could be modified in order to achieve a more adaptive- and expressive-oriented general definition, with the ultimate aim of improving the specification and bringing it closer to actual needs on eLearning.

In this section, we provide a detailed analysis of what IMS-LD can and cannot model, in its current information model, with regards to adaptation. This analysis concentrates on the weak points and main features of every learning scenario. These remarks will be addressed to produce a set of recommendations (i.e. extensions and modifications) to improve the pedagogical expressiveness on IMS-LD, focused on adaptation, in the next section.

Following, we summarize our main findings. With regards to the specification itself:

1. The definition of properties and the link through several working XML files is too complicated to become useful
2. The relation between layers and actions is not straightforward and it has to be done interlacing files, through global elements and XML

[2] www.coppercore.org

3. The lack of a richer conditional structure makes the editing of the set of rules more complicated on paper than they actually are from a rational point of view

4. Controlled iterations in the activities are not allowed. Furthermore, a closed activity cannot be re-initialized and/or go backwards

5. The monitoring service doesn´t cover any kind of user grouping. Therefore, a user (e.g. either a teacher or a learner) cannot follow the performance of several other users at the same time

6. Questions and answers are not personalised for user; they are identical for all users with the same role

7. The communication between teacher and student is little and indirect. They can view the values of properties but there is no other communication service between them

8. There is a lack of flexibility in the input point of changing the itineraries. In the type Sequence, the learning activity with the question appears always at the same place. In the type Selection, the question is always presented after 2 completed learning activities. In case the learning designer/teacher wants to shift this input point, they cannot do so

9. There is no possibility to handle absolute time to start the course and/or a specific activity. Only relative time to the precise time when the instance is created out of the UoL, it is possible

10. There is no chance to make a connection to an already existing database (for instance, to make a query or to import already enrolled students or teachers). The data type of connection is not supported. Therefore, every enrolment has to be done by hand or running a specific tool for that

11. Furthermore, any connection with the external world is impossible. For instance, a real-time effective communication between an LMS and an IMS-LD UoL is not possible so far, so that in fact they cannot benefit each other from mutual services and resources. There is no foreseen dispatcher or service in the specification allowing such connection [Moreno et al., 2007]

12. When an executable module is developed with other technologies (Macromedia Flash and PHP, for instance), it cannot be integrated with IMS-LD in any way. Therefore, we also identified an interoperability problem. Although IMS-LD is not developed with the intention of supporting such interactivity with users, it could allow for a valid integration with external resources using a layer of communication/dispatcher.

13. A file uploaded from the hard disk of a computer is stored in a file-type property inside the internal database of the engine (CopperCore, in this case). There is no possibility to change the default configuration for storing or retrieving resources. There is no facility to manage those uploads either. Although this is an issue concerning tools too, the core documents of IMS-LD do not provide this information and/or service either

14. IMS-LD does not allow saving information into external files or retrieving information from any external source

15. To perform a dynamic user selection in order to create groups is not possible. The teacher can monitor each user, and provide him/her with some feedback on a personal basis. We could set-up a property to be dealt by groups, but these groups should be established before the actual start. However, if the teacher wants to make a dynamic creation of a group of students depending on their answers, this is not possible so far. To this extent, groups and roles are the same thing

16. IMS-LD does not allow for recording the user's behaviour; in fact, no measures (i.e., Total Time Needed, Time Before First Move) can be restored or retrieved

17. As a consequence, adaptation based on the user´s behaviour cannot be developed using the IMS-LD specification. Furthermore, the current state of tooling does not support it either

In addition, with regards with the current engines, we highlight a few issues that would support a more powerful use of the specification:

18. Changes on-the-fly are not possible. In case that the teacher or the learning designer wants to change i.e. the questions, the answers, or the content of the next activity to be carried out, they find that. Every single resource has to be packed in design and publishing time before the actual running of the instance

19. In questionnaires and other forms with fields, the teacher/learning designer cannot modify the number of questions or answers, once the UoL has started

20. There is no option to run the UoL (the whole UoL or a part, such a Learning Activity) twice within the same instance. Once a Learning Activity is closed, the user can read it again but the associated learning flow cannot be executed. For instance, after the question to change the itinerary is made in the historic-route, there is no way to go back

21. There is no flexibility to change the content. When the teacher/learning designer wants to keep the same method and the same structure, but he/she wants to change one single HTML page with some content, the UoL has to be validated and published again. In this case, the learner and the teacher would have to be enrolled and the learning process starts from the very beginning

22. Users cannot be dynamically enrolled within the UoL, once it has started, and they have to be managed by an external tool

6 Further analysis

In the next section we show specific recommendations which deal with extensions, modifications of modelling structures, elements and components, as well as with the architecture of IMS-LD. Those recommendations are based on the constraints pointed out in Section 5. However, there is a need for presenting some further analysis, which can bridge both sections, from the constraints to the recommendations, since this in-between step is crucial to understand the rationale. We have organized the analysis as follows:

a) Analysis on general-purpose modelling. These elements will be used as part of others specifically implemented in learning processes, like personalisation. Furthermore, they become a basic set to be re-purposed in different contexts and goals. Therefore, this initial analysis comprises adaptive learning. A few very specific processes cannot be approached with just general structures. They need on-purpose elements which come across on-purpose goals on personalisation

b) Analysis on the integration of Units of Learning and a bi-directional communication with other external resources, systems and standards. When needed, we highlight the need for a way of communication (e.g., a communication layer) although its development is something outside of the scope of this research. We are focused on

the specification itself and how to improve the pedagogical expressiveness, and not on building any *ad hoc* technical artefact to get this aim through.

Out of this analysis, we conclude that specific recommendations should be categorized in three groups:

a) Modelling, that compiles every single extension, modification or addition, general or specific, to the specification and the information model; and b) Architecture, that deals with functional requirements of the spec, with a focus on the interoperability, communication and integration of IMS-LD with other external means. In both cases, we look for the highest performance along with the minimal structural change. Furthermore, we respect the original specification as much as possible and try to make as few changes as possible; on the other side, they all are needed to build the suggested solution, and cope with the overall approach. In addition, c) we reflect some recommendations about the authoring tools. Although they are not responsibility of the specification, they are indeed related to IMS-LD, since the tools which allow the end users to create useful and applicable Units of Learning, can make the process easier or more difficult, and therefore it constraint the actual use and outcomes.

Furthermore, we depict our conclusions within the same two main blocks that we have used to carry out the analysis: modelling (with a special focus on adaptation) and integration. Out of our solution, we also provide a brief note about authoring tools.

6.1 Modelling and Adaptation

With regards to general modelling, and modelling focused on adaptive learning we conclude that IMS-LD shows a metaphor difficult to understand. It is not as much to say that people do not understand what a theatre is or how a play is performed. The key issue comes when a teacher needs to translate this well-know structure into specific pedagogical resources and features. This translation process turns not to be so obvious. The conceptual model is clear: play, acts, roles, role-parts, and so on. But all of them, interlaced in a whole structure of learning, become complex. Even the simplest scenario requires some knowledge of the specification in a technical way. And this is far from being user-friendly, moreover when the usual target people consists of non-technical profiles.

The notation itself follows a usual XML Schema and the definition of the several elements and components of the spec can turn too complex, even for skilled programmers. The description of activities, activity structures, environments, and etcetera, and the long cascade of relationships amongst them, makes a difficult-to-trace chain out of a simple scenario. Not to mention when several roles are involved, when some components of Level B are used or when adaptive processes are required. The programming structure is quite easy, but the combination of elements, components and metaphor, makes it hard difficult to implement.

The programming components provided by IMS-LD are quite simple (i.e., simple condition, based arithmetic, visualization of variables, visibility, DIV layers, and etcetera). On the other side, their syntax is long, which hinders the rationale of the modelling process itself.

6.2 Communication, interoperability, integration of Units of Learning

We study three ways of communication: 1) simple link between parts, 2) embedded information packages with no information exchange, and 3) full communication of information packages, sharing variables and states. This third solution becomes the most effective one. It implies the development of a communication layer that deals with effective bi-directional exchange of data between information packages. Furthermore, this solution allows for the communication and sharing of services, along with variables, values and states, between IMS-LD and any outside counterpart, i.e., other specifications (e.g. SCORM), languages (i.e. PHP, Java, and Action Script), and LMSs (i.e. LAMS[3], Moodle[4], .LRN[5]).

Should this exchange actually happens, it will encourage the re-use of information packages in different contexts, and the development of templates, fostering the repurpose of Units of Learning within and amongst the several communities of practice (target groups) involved in IMS-LD, beyond the very only technical niche.

In the same line, exportation and importation of Units of Learning is not developed so far; neither does any connection with a database. Once more, no information exchange with other entities is possible so far.

The current two-step working process that makes two isolated parts out of design-time and run-time, makes IMS-LD to be compiled and not interpreted. This distinction stops an on-the-fly visualisation and modification of the learning design, which would improve the interactive personalisation of the learning process. This issue deals with how IMS-LD is interpreted by tools and engines developers and not with how the specification is actually designed.

6.3 Authoring

As aforementioned, this research and paper are focused on the specification itself and it does not deal with tools. However, authoring tools largely influence what can be modelled and how. Therefore, we point out a couple of key issues that could support the actual adoption of IMS-LD by the target groups:

a) There is a need for high-level visual authoring tools. Nowadays there are two types of tools: effective but too technical, even for technical profiles; and simple to understand but not powerful, since they usually deal with the very basic Level A. The creation of UoLs should be as far as possible from technical requirements or the underlying elements, components or structure. A more visual approach would encourage the understanding and use of IMS-LD in a broader sense by target groups. Technical low-level editors should live along with the visual high-level ones, though

b) Any authoring tool should allow for an integrated modelling, working with the manifest, the resources and the required external XHTML files with a common interface. It should dependencies and ease setting of properties. This is a hot challenge, not possible so far.

[3] www.lamsfoundation.org
[4] http://moodle.org
[5] www.dotlrn.org

7 Recommendations: Extensions and modifications

This section presents a rich and structured set of recommendations, modifications and extensions to improve the expressiveness of IMS Learning Design on adaptive learning processes. It lays on the aforementioned analysis. The following set of tables show a summary of the constraints, analysis, and recommendations (Table 2). The tables are structured as follows: in the grey-coloured, first row of each table, Column 1 (ID) numbers the constraints and analysis issues. Prefix M relates to issues concerning Modelling, and prefix A relates to issues concerning Architecture. Column 2 (Constraints...) provides a description of those issues numbered in Column 1. The white-coloured row(s) afterwards, presents the recommendation/s in the same couple format: ID and description.

ID	Constraints, analysis and recommendations
[M.01]	Programming structures and resources are very basic (simple condition, simple arithmetic, properties set-up, visibility, DIV layers)
[Rec.01a]	Condition type case
[Rec.01b]	Condition type case with automatic ranges
[Rec.01c]	Conditional loop, type while
[Rec.01d]	Integer loop, type for-next
[Rec.01d]	Modification of the element <calculate>

[M.02]	There is no management of absolute time. There is no synchronization nor input point to work with relative time from
[Rec.02]	Modification of reference to relative time. Addition of reference to absolute time

[M.03]	Notification service, in Level C, is underused. It only sends an email or plays an activity
[Rec.03]	Extension of the notification service, beyond using *sendmail* and playing an activity. It can be called from other structures besides the <on-completion> part of a learning activity

[M.04]	There is a blur way to handle the definition and use of properties and links amongst the several XML with global elements
[Rec.04]	Syntax modification, definition and use of elements view-property and set-property, as long as the properties which make use of them

[M.05]	Relationship between DIV layers and the visibility property is difficult to make and follow
[Rec.05]	In principie, the visibility property of any layer is turn off (hide), making simpler the condicional structure which could make use of it

[M.06]	There is no chance for iterations in any of the basic structures of the IMS-LD metaphor (learning activity, support activity, activity structure, act, play)
[Rec.06]	Extension of the current syntax of every element with a parameter <iteration> which defines a integer loop (type for-next) and-or a conditional loop (type while)

[M.07]	There is no synchronization input point in the manifest
[Rec.07]	Addition of an element GOTO which allows for a direct guiding of the learning flow

[M.08]	There is no chance to assign a specific activity to a selected user
[Rec.08a]	Addition of an element ASSIGN-ACTIVITY-TO-USER which allows for a direct match amongst users, groups and roles, with learning activities and activity structures
[Rec.08b]	Addition of an element ASSIGN-USER-TO-ACTIVITY which allows for a direct match amongst users, groups and roles, with learning activities and activity structures
[Rec.08c]	Addition of an element SWITCH-ACTIVITY which allows for turning on-off activities and activity structures

[M.09]	There is no chance to make groups out of a selection inside the instance
[Rec.09]	Addition of an element CREATE-GROUP which allows for grouping users of the same role

[M.10]	The monitoring service does not allow for monitoring of groups
[Rec.10]	Extension of the monitoring service to trace roles and groups

[A.11]	IMS-LD does not allow for saving or retrieving data in external files, of any kind of format. In addition, connections with external databases or modules developed with other languages are not described or supported within the specification
[Rec.11a]	Addition of the elements EXPORT and IMPORT to handle files with specific parameters (e.g., type TXT) and which is defined in a new property type FILE-IO
[Rec.11b]	Addition of an the elements FROM-DB and TO-DB which allows for saving and retrieving data in a database of type MySQL. The connection is defined in a new property type DATABASE

[A.12]	There is no chance to modify the learning skeleton, method, roles definition or any other structural element in run-time
[Rec.12]	Addition of two couples of global elements: a) *view-IMS-LD* y *set-IMS-LD*, b) *view-resources* y *set-resources*, which allows for the visualisation and modification

	of the learning design and the related resources in run-time

Table 2. Constraints, analysis and recommendations

At the project website pointed out in Section 3, every recommendation is expressed in an XML format, along with a full description, and one example. For instance (Figure 2):

```
<calculate>
<!-- "ID-OP-1" * ("ID-OP-2" + 3) -->
   <property-ref ref="ID-OP-1" />
   <multiply>
   <group-subtotal ref="SUB-1">
      <property-ref ref="ID-OP-2" />
      <sum>
         <property-value>3</property-value>
      </sum>
   </group-subtotal>
   </multiply>
</calculate>
```

and

```
<learning-activity isvisible="true" identifier="activity-1">
      <title>Activity to carry out</title>
      <activity-description>
         <title>First part of the activity</title>
         <item isvisible="true" identifierref="item-1"/>
      </activity-description>
      <iteration>
        <is>
          <property-ref ref="Answer1"/>
          <property-value>A</property-value>
        </is>
      </iteration>
</learning-activity>
```

Figure 2. Example snippets of two recommendations

8 Conclusions and further work

This paper shows the background about IMS Learning Design and how to model adaptive learning with this specification. In addition, we provide a thorough analysis of a number of learning scenarios and a detailed list of issues to be modified and improved in the specification to better express adaptation. Based on these outcomes we provide recommendations, modifications and extensions to IMS Learning Design in order to improve its expressiveness of adaptive learning.

With these regards, Level A of IMS-LD provides the basic skeleton and a general framework to work with Units of Learning. It makes the 80% of the whole structure. Level C, and above all Level B provide both the spec with stronger and more versatile resources. These two upper levels are the actual responsible means to model some of the current learning and teaching challenges (i.e., active learning, collaborative learning, adaptive learning, runtime tracking).

Furthermore, we examine how to represent adaptive and adaptable Units of Learning with IMS Learning Design in order to model different types of Adaptation. Based on a literature study, a distinction is drawn between eight types of Adaptation that can be classified in two clusters: a) the main group, with interfaced-base, learning-flow and content-base; b) interactive problem solving support, adaptive information filtering, adaptive user grouping, adaptive evaluation, and changes on-the-fly. Out of this research and modelling efforts we derived a number of findings focused on the limitations that IMS-LD provides. These findings are mainly focused on adaptive learning process. However, since this topic cannot be isolated from the overall approach of the specifications, some of the limitations, and further recommendations, also address other topics, like interoperability, or even authoring tools.

Indeed, IMS-LD will benefit from a re-structure and modification of several elements focused on modelling and architecture. It will also improve the overall pedagogical expressiveness, along with specific features on adaptation of learning processes and integration with other specifications, LMSs, and learning resources. These are two main objectives of the specification: personalised learning and interoperability. At the same time, IMS-LD would increase its level of implementation in real settings and a wider support from Communities of Practice of end users if one or several high-level visual authoring tools are developed. Nevertheless, this issue is out of the scope of this research, and it deals with research groups and companies working on the adoption of IMS-LD.

Acknowledgments

The research presented in this paper has been partially supported by the following projects: FLEXO (Spanish Plan Avanza, www.ines.org.es/flexo, TSI-020301-2009-9), GRAPPLE (FP7-ICT-2007-1, www.grapple-project.org, contract number 215434).

References

[Ahmad et al., 2004] Ahmad, A., Basir, O., and Hassanein, K. Adaptive user interfaces for intelligent e-Learning: issues and trends. Proceedings of The Fourth International Conference on Electronic Business, ICEB2004, Beijing, 2004

[Baeza-Yates & Ribeiro-Nieto, 1999] Baeza-Yates, R., and Ribeiro-Nieto, B. Modern Information Retrieval. Boston, MA, USA: Addison-Wesley, 1999

[Brusilovsky & Miller, 2001] Brusilovsky, P., and Miller, P. Course Delivery Systems for the Virtual University. In Tschang F.T. & T. Della Senta (Eds.), Access to Knowledge: New Information Technologies and the Emergence of the Virtual University (pp. 167-206). Amsterdam: Elsevier Science and International Association of Universities, 2001

[Burgos & Griffiths, 2005] Burgos, D., and Griffiths, D. The UNFOLD Project. Understanding and using Learning Design, 2005. Available at http://hdl.handle.net/1820/548

[Burgos & Specht, 2006] Burgos, D. and Specht, M. Adaptive e-learning methods and IMS Learning Design. An integrated approach, Proc. ICALT2006, 2006. Available at http://www.ask.iti.gr/icalt/2006

[Burgos et al., 2007] Burgos, D., Tattersall, C., and Koper, R. How to represent adaptation in eLearning with IMS Learning Design. Interactive Learning Environments, 15(2), 161-170, 2007

[Burgos, 2008] Burgos, D. Extension of the IMS Learning Design Specification based on Adaptation and Integration of Units of Learning (Extensión de la especificación IMS Learning Design desde la Adaptación y la Integración de Unidades de Aprendizaje). Doctoral thesis. University Carlos III, Leganés, Madrid, Spain, 2008

[Butz et al., 2003] Butz, M. V., Olivier, S., and Gérard, P. Anticipatory Behavior in Adaptive Learning Systems : Foundations, Theories, and Systems. In. Berlin: Springer Verlag, 2003

[Chin, 2001] Chin, D. Empirical Evaluation of User Models and User-Adapted Systems. User Modeling and User-Adapted Interaction, 11, 181-194, 2001

[De Bra et al., 2004] De Bra, P., Aroyo, L., and Cristea, A. Adaptive Web-based Educational Hypermedia. In M. Levene & A. Poulovassilis (Eds.), Web Dynamics, Adaptive to Change in Content, Size, Topology and Use (pp. 387-410): Springer, 2004

[IMSCP, 2001] IMS. IMS Content Packaging Specification. Version 1.1.3, 2001. Retrieved April 28th, 2009, from http://www.imsglobal.org/learningdesign/index.cfm

[IMSLD, 2003] IMS. IMS Learning Design. Version 1. Retrieved February 27th, 2004, from http://www.imsglobal.org/learningdesign/index.cfm

[Koper & Burgos, 2005] Koper, R., and Burgos, D. Developing advanced units of learning using IMS Learning Design level B. International Journal on Advanced Technology for Learning, 2(4), 252-259, 2005

[Koper & Tattersall, 2005] Koper, R., and Tattersall, C. Learning Design: A Handbook on Modelling and Delivering Networked Education and Training. Berlin Heidelberg: Springer, 2005

[Merceron & Yacef, 2003] Merceron, A., and Yacef, K. A Web-based tutoring tool with mining facilities to improve learning and teaching. Proceedings of AI-Ed'2003: IOS Press, 2003

[Moreno et al., 2007] Moreno-Ger, P., Burgos, D., Sierra, J. L., & Fernández-Manjón, B. (2007). An eLearning specification meets a game: authoring and integration with IMS Learning Design and <e-Adventure>. Proceedings of ISAGA. 2nd international workshop on Electronic Games and Personalized eLearning Processes (EGAEL2007). July, 9th-13th, 2007, Nijmegen, The Netherlands

[Romero et al., 2003] Romero, C., Ventura, S., De Bra, P. D., and Castro, C. D. (2003). Discovering prediction rules in AHA! courses. Proceedings of 9th International User Modeling Conference (pp. 25-34), 2003

[Specht & Burgos, 2006] Specht, M., and Burgos, D. Implementing Adaptive Educational Methods with IMS Learning Design. Proceedings of Adaptive Hypermedia 2006, Dublin, Ireland, 2006

[Van Rosmalen & Boticario, 2005] Van Rosmalen, P., and Boticario, J. Using Learning Design to support design- and runtime adaptation. In R. Koper & C. Tattersall (Eds.), Learning Design: A Handbook on Modeling and Delivering Networked Education and Training. Heidelberg, Germany: Springer Verlag, 2005

Student Model Adjustment Through Random-Restart Hill Climbing

Ahmad Salim Doost
Saarland University
66123 Saarbruecken, Germany

Erica Melis
German Research Center for Articial Intelligence (DFKI)
Stuhlsatzenhausweg 3, 66123 Saarbruecken, Germany

Abstract

ACTIVEMATH is a web-based intelligent tutoring system (ITS) for studying mathematics. Its course generator, which assembles content to personalized books, strongly depends on the underlying student model. Therefore, a student model is important to make an ITS adaptive. The more accurate it is, the better could be the adaptation. Here we present which parameters can be optimized and how they can be optimized in an efficient and affordable manner. This methodology can be generalized beyond ACTIVEMATH's student model. We also present our results for the optimization based on two sets of log data. Our optimization method is based on random-restart hill climbing and it considerably improved the student model's accuracy.

1 Preliminaries

An intelligent tutoring system (ITS) is a computer system that provides personalized support (either by giving feedback or instruction) to students performing tasks without the intervention of human tutors. They can improve education quality by providing new possibilities (e.g. for homework assignments or self-study opportunities). Therefore, ITSs can be used to support teaching and diminish the teacher shortage problem [Flynt and Morton, 2009; Ingersoll and Perda, 2009] to some extent. ITSs can be permanently available, widely reusable, location-independent and adaptive. Adaptivity is an important feature of ITSs: it allows to focus the education to pedagogically useful content like teaching content the student has difficulties with and keeping the student from working on too easy or too difficult exercises. To make ITSs able to adapt to single students, an underlying student model aiming to represent certain aspects of a student (most importantly his/her skills), is required.

A student model represents variables and characteristics of a learner which are relevant for educational purposes. A typical learner variable is the student's state of knowledge concerning various concepts and competencies. This is continuously adjusted with the learner's progress over time. When developing a student model several parameters have to be determined which, in a first run, may be defined in an ad hoc way. However, these values are usually not appropriate and also wrong interrelations may cause problems. Instead, an evaluated configuration might improve the accuracy of the student model, which in turn leads to a better individualized learning experience.

How can we optimize parameters in a generic way? First of all, the parameters of ACTIVEMATH's student model, also called *Salient Learner Model* (SLM), which we want to optimize are described. We demonstrate how we adapted the Random-Restart Hill Climbing (RRHC) algorithm [Russel and Norvig, 2003] and used it to adjust SLM. Finally, we provide evaluation results and describe how the methodology can be generalized beyond SLM.

1.1 ACTIVEMATH and SLM

ACTIVEMATH is a web-based ITS for studying mathematics[1] [Ullrich and Libbrecht, 2008]. It is being developed at Saarland University and at the German Research Center of Artificial Intelligence (DFKI) since the year 2000. Its learning content (like definitions, examples, exercises and explanations), are semantically encoded in an extended OMDoc [Kohlhase, 2006]. The OMDoc documents are then converted by the presentation component to the desired output format (like HTML or PDF, cf. Figure 1).

Students cannot only browse through prerecorded static books. With the course generator of ACTIVEMATH, learning objects can be organized into a book which is specific to the student's abilities: The generated book depends on the student's goals and on the modeled belief about the learner's competencies. The goal can be for example to discover a new topic, revise an already known topic or simulate an exam. The content is selected according to special rules of the course generator and depends on the knowledge of the learner. The ACTIVEMATH course generator aims to generate user-specific courses with relevant learning material and appropriate difficulty. For example, if the student wants to discover a new topic but his competencies on some of the prerequisites for the topic are poor, then the course generator can include them in the personalized book with an easy level of difficulty. The content of the personalized dynamic books are updated and restructured with the student's progress over time.

The exercise system of ACTIVEMATH runs interactive problems with personalized feedback. Like other modules in ACTIVEMATH, the exercise system publishes events to notify the remaining parts of the system about the user interactions. The student model of ACTIVEMATH [Faulhaber and Melis, 2008] listens to these events and updates its belief after every exercise step. It takes the learner's achievements into account together with the exercise metadata (e.g. exercise difficulty).

The most relevant metadata is the set of trained concepts and cognitive processes. *Addition of fractions* or *Pythagorean theorem* are examples for concepts. Concepts

[1] `http://demo.activemath.org`

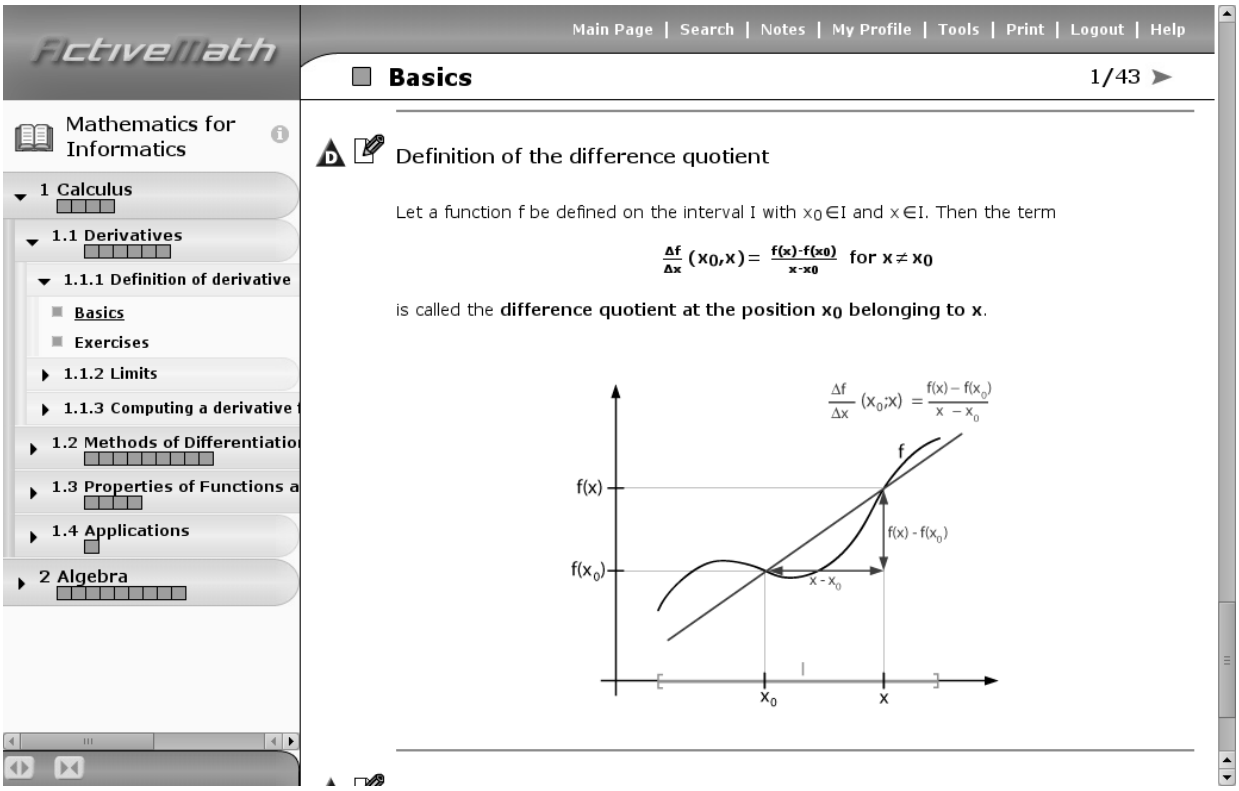


Figure 1: A page of a book in ACTIVEMATH from a web-browser.

can be connected with each other with relations like the *prerequisite*-relation specifying that in order to be able to master one concept, the prerequisite-concept has to be mastered first (e.g. the concept *adding fractions with equal denominator* forms a prerequisite of the concept *adding fractions with unlike denominators*). The cognitive process metadata defines what process (like *detecting errors* or *applying algorithms*) is trained. We refer to the pair of a concept and a cognitive process as *competency*. For example, *detecting errors about the Pythagorean theorem* is a competency. Another very important metadata that exercises are required to be annotated with, is one of the five difficulty values: `very easy`, `easy`, `medium`, `difficult` or `very difficult`.

Out of a set of evidences, consisting of a concept, a cognitive process, a difficulty value and the achievement (whether the exercise was solved correctly), the student model estimates students' proficiencies. These evidences are generated after each approach to a problem. For each pair of a concept and a cognitive process (i.e. for each *competency*) defined in the metadata of an exercise, evidences are generated, propagated to related concepts (thus becoming *indirect evidences*) and stored in its specific containers. The modeled belief about the student's competency is then updated based on the available evidences, where indirect evidences have less influence on the resulting mastery value. Figure 2 illustrates this process.

1.2 Item Response Theory in SLM

The calculation of the probabilities for different masteries is based on the Item Response Theory (IRT) [Lord, 1980], which describes the relationship between the probability for a correct answer X to a mastery value (ability) θ:

$$P_i(X = correct|\theta) = c_i + \frac{1 - c_i}{1 + e^{-a_i \cdot (\theta - b_i)}} \qquad (1)$$

The resulting function is called the *item characteristic curve* (ICC). Equation 1 shows the three-parameter model of the item response theory. The parameters are exercise (item) specific where

- a_i is the item *discrimination factor*,
- b_i is the item *difficulty*,
- $c_i \in [0, 1]$ is the item *guessing probability*.

The item difficulty b_i defines the location of the mastery value θ where the probability of correct response is exactly 0.5, i.e. $P_i(X = correct|\theta = b_i) = 0.5$. Hence, the larger b_i is, the larger the mastery θ has to be to have a probability of at least 50%.

The discrimination factor defines the steepness of the *S*-shaped curve. Therefore, it describes how well it is differentiated between learners having masteries below the item difficulty ($\theta < b_i$) and those having masteries above the item difficulty ($\theta > b_i$). Figure 3 shows different ICCs with differing item difficulties.

The actual domain of the mastery θ, the item discrimination factor a_i, and the item difficulty (or item location) b_i is $\mathbb{R}$. However, if one restricts the domain of the item difficulty b_i, then the most differing probabilities for different mastery values θ will be around that range as well. This is because of the nice property that $P_i(X = correct|\theta = b_i) = 50\%$. SLM makes use of this and restricts the domain of θ and b_i to $[0.0, 1.0]$, which simplifies the interpretation of difficulty values.

Furthermore, SLM restricts its calculation to those masteries θ which are "near" the item difficulty (i.e. $\theta \in [b_i - \delta, b_i + \delta]$, where δ is the *information radius*). This will keep the student model from assuming high mastery values for a student who solved easy exercises only.

For more details on how the student model of ACTIVEMATH works, please refer to [Faulhaber, 2007; Faulhaber

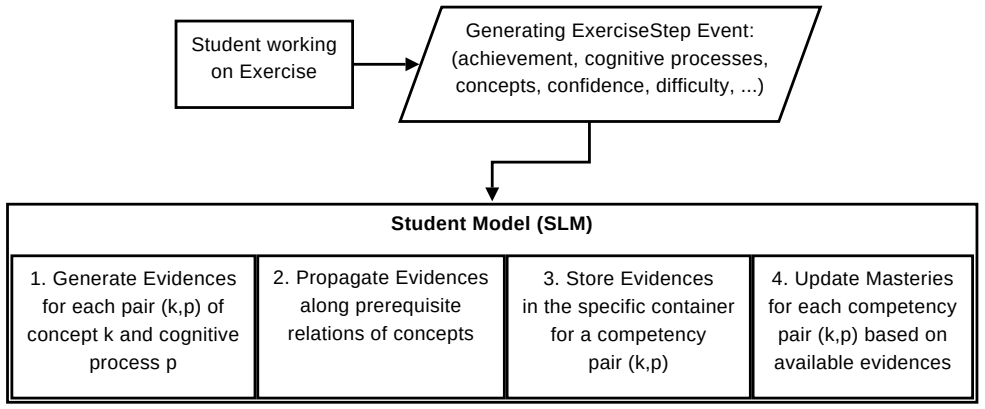


Figure 2: The interaction of a student with ActiveMath creates events, which are handled by the student model.

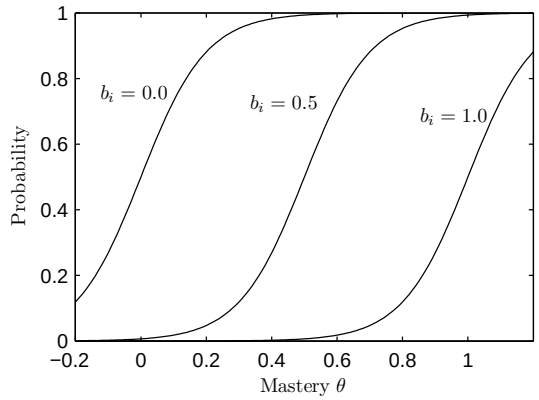


Figure 3: Typical ICCs showing probabilities of a correct answer for different masteries θ and varying item difficulties b_i. The guessing probability c_i is fixed to 0 and a_i to 10.

and Melis, 2008; Doost, 2010].

2 Which SLM Parameters can be Adjusted?

Depending on the structure, design and implementation of a student model, different ways of adjustments and optimization possibilities emerge. This section presents most of the student model parameters that have been selected for the optimization process. Some of the parameters were actually hard coded into the system. In order to discover them, a thorough analysis of the implementation was necessary.

Propagation Depth defines the maximum distance an evidence is propagated along the graph of prerequisites for concept relations. Too deeply propagated evidences might cause the SLM to transfer the student's knowledge about one concept to other possibly too little related concepts. However, with a completely deactivated propagation we might lose possible accuracy gains for concepts for which direct evidences are not available yet.

Weighing Evidences. How strong shall we weigh direct evidences compared to indirect ones when both are available? Giving no weight to indirect evidences is equal to having no propagation. However, giving them too much weight might understate the significance of direct evidences.

Evidence Container Size is the maximum number of stored direct and indirect evidences per competency and represents the *forgetting factor*. When this number is reached, old evidences are discarded in favor of new ones.

Difficulty Mapping. What is the range $[\alpha, \beta]$ into which the five difficulty values are mapped (every exercise in ACTIVEMATH is annotated with one out of five possible values). Probabilities received from the IRT strongly depend on the difficulty of an exercise. An arbitrary mapping might lead to misinterpretations of difficulty metadata and negatively effect the mastery estimations.

Default Discrimination Factor determines the curvature of the sigmoid function, which converges to a step function for high values making marginal masteries more probable and mid-level masteries less probable.

3 Random-Restart Hill Climbing

Hill climbing [Russel and Norvig, 2003] is a greedy local search algorithm and can be used for optimization problems. Hill climbing algorithms can find reasonable solutions in large or infinite (continuous) state spaces for which systematic algorithms fail.

Generally, when looking for a maximum of a function (optimization problem), the hill climbing algorithm works as follows:

1. Start at an arbitrary point
2. Calculate values for neighboring points
3. Move to the point with increased value
4. Terminate if no higher value could be found, otherwise continue at 1

The standard problem with this algorithm is that it may not find the optimal solution (i.e. the global maximum), but only a local maximum. However, with an extension known as random-restart one can increase the probability to find a global maximum considerably [Russel and Norvig, 2003]. Starting the hill climbing algorithm over and over again each time with randomly chosen initial states and saving only the maximum of the new values improves the probability of finding the global maximum.

The complex structure of the student model and the possibly interrelated parameters make it difficult - if not practically impossible - to determine the optimal values for the

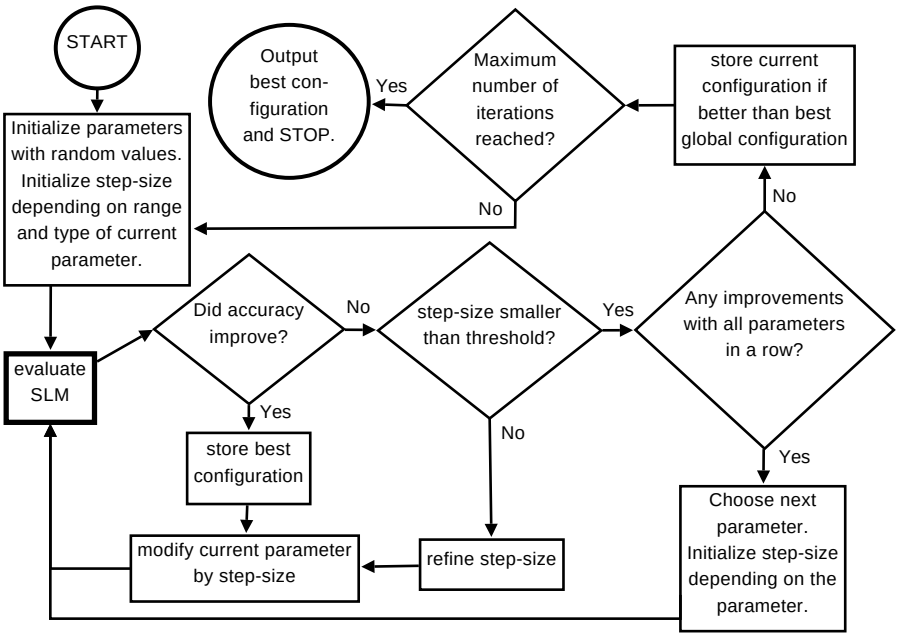


Figure 4: A flowchart of the modified RRHC algorithm used for parameter optimization.

parameters by classic machine learning methods. For optimizing the parameters mentioned in Sect. 2, the Random-Restart Hill Climbing (RRHC) algorithm has been modified. Our version steadily switches between the several different parameters. It modifies each parameter as long as improvements could be found, then switches to the next and returns to that parameter again as soon as all other parameters were taken into account. We restart with a new initial random state if no parameter change yielded improvements and terminate after a predefined maximal number of iterations.

The continuous state space of parameters with real values requires additional attention. In order to find the maximum quickly, we need to make noticeable larger steps than the smallest possible difference. This is done by using an initial step size δ depending on the range size of possible values for a parameter. The larger the interval, the larger will be δ. The parameter is increased by δ as long as there is any improvement. Then, the step size is refined by a factor $0 < \alpha < 1$ until the step size gets smaller than a predefined threshold ε (we used 0.1 for α and 0.0001 for ε). We continue with negative step sizes starting from $-\delta \cdot \alpha$ and increase it by multiplying with α again until we reached a value larger than $-\varepsilon$.

In short, we first start with a rough search by modifying a parameter with larger steps, then we conclude with fine tuning by modifying a parameter with smaller and smaller steps until we reach a certain threshold for a minimum step size. A flowchart for our modified version of RRHC is presented in Figure 4.

4 Evaluation and Results

An important question remains: what is the best function for the optimization problem in order to find a good parameter configuration for a student model?

The student model can be evaluated by replaying it with empirical data collected during another experiment. Based on this data an experiment can be simulated for each possible configuration without organizing new ones. Its log data can be rerun by going through all event entries chronologically and passing them over to the student model. During the simulation process we can query the student model about its belief and compare it to the actual achievements before passing on this information to the student model. Figure 5 depicts a diagram of this evaluation process.

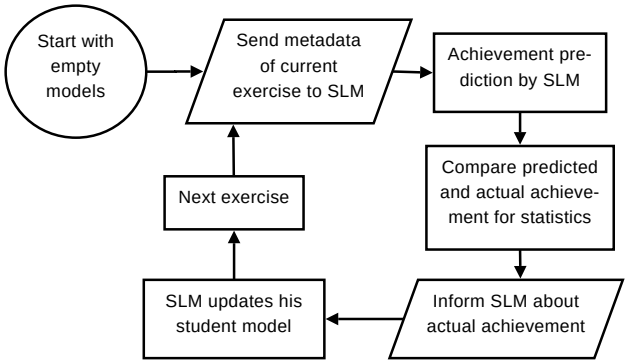


Figure 5: The evaluation process of the student model SLM through replay of log data.

We believe that the more accurate the student model is, the better are the *achievement* predictions, i.e. whether a student is able to solve a specific exercise or not. Our function for the optimization problem is therefore the achievement prediction accuracy which we strive to maximize. It is calculated as shown in Figure 6, where n is the overall number of exercise steps available (i.e. the total number of predictions), ach_i is the actual achievement in the ith exercise step[2], and $pred_i$ is the student model's prediction for the achievement of the ith exercise step.

For the evaluation, log data collected in the context of the ATuF project [Eichelmann *et al.*, 2008] (with 190 students, 6th and 7th graders) and data from experiments of the ALoE project [3] [Tsovaltzi *et al.*, 2009] (with 77 students, mostly 6th graders) has been used.

[2]It is 1 if the student's answer was correct, otherwise 0.

[3]Funded by the German National Science Foundation (DFG) (ME1136/7).

$$accuracy = \frac{\text{number of correct predictions}}{\text{overall number of predictions}} = \frac{1}{n} \cdot \sum_{i=1}^{n} 1 - |ach_i - pred_i|$$

Figure 6: Achievement Prediction Accuracy

4.1 Results

For the ATuF data, the original SLM had an achievement prediction accuracy of 79%. This is already a highly improved and optimized value, which was the result of several years of experimentation. Still, with the parameter configuration obtained from the presented method, the accuracy became 82%. In absolute numbers, this means that 470 out of 15680 predictions which were wrongly predicted before are now predicted correctly. For content selection based on the estimated competencies, every wrongly selected item (like selecting too easy or too difficult exercises, or leaving missing prerequisites away) could make considerable differences.

A statistical hypothesis test (one-sided) with a statistical significance of 5% and the null hypothesis being the non-improvement of the prediction accuracy yields a z-score of -6.3. This means that we have to reject the null hypothesis (since the z-score is less than -1.65) and assume that the alternative hypothesis is true, namely that the accuracy improvement by 3% is *(strongly)* significant. The evaluation for the second data set, the ALoE data, revealed for the original SLM an overall prediction accuracy of 69%[4]. With the optimized parameter configuration the overall prediction accuracy rose by 2% to 71%. This yields a z-score of 2.1, which is statistically significant as well.

In the following, the single parameter values of the configuration which yielded the highest accuracy will be mentioned briefly. Surprisingly, a *propagation depth* of one produced almost the same accuracy as no propagation. However, a depth of two or more worsened the result. This would mean that we cannot really transfer the students' knowledge level of one concept to related concepts except to directly adjacent ones (without any gain, though). For *weighing* direct and indirect evidences a ratio of $12:1$ was obtained. As expected, the value of direct evidences is considerably higher. Furthermore, keeping a maximum of six *evidences* per competency produced the highest accuracy. So did the interval $[0.14, 0.86]$ for the *difficulty mapping*, which means that the easiest exercises get a difficulty value of 0.14 instead of 0. Finally, a *discrimination factor* of 9.6 showed to be best.

5 Related Work

Finding optimal parameter configurations is a problem many student models have to tackle. The parameters may belong to Bayesian networks, to metadata of exercises or concepts (also known as rules), to the working mechanism of a student model itself or to any other part related to the student model. Corbett and Anderson conducted several experiments to evaluate their knowledge tracing student model. From the resulting data they fit the weights for learning and performance parameters of each rule and student [Corbett and Anderson, 1995]. Cen et al. have shown that using an educational data mining technique called *Learning Factors Analysis* (LFA) they could improve the learning efficiency by 12% [Cen *et al.*, 2007]. For this, they optimized knowledge tracing parameters of the Cognitive Geometry Tutor based on older log data. In the experiment with the optimized model students required on average 12% less time to reach the same level.

Gong, Beck and Heffernan compared two techniques for student modeling, *Knowledge Tracing* and *Performance Factor Analysis* (PFA), in terms of their predictive accuracy and parameter plausibility [Gong *et al.*, 2010]. For fitting the knowledge tracing model, they used and compared two different techniques: *expectation maximization* and *brute force*. To minimize the problem of local maxima, they restart their fitting algorithm multiple times as well. In their work, Gong et al reported that the *brute force* method was not as good as the *expectation maximization* method. *Brute force* is a very expensive and – because of the continuous state space – an infeasible model fitting method. Random-restart hill climbing (RRHC) is a method in between *expectation maximization* and *brute force*. In a brute force manner it evaluates and tries out different parameter configurations. However, it does not evaluate an equally distributed sampled set of configurations, but instead it modifies parameters stepwise in the direction of expected improvement until a maximum is reached.

Local search methods, like hill climbing, are often used for finding solutions efficiently, especially for NP-hard problems like the traveling salesman problem [Aarts and Lenstra, 2003]. Random-restart or multiple-restart is a known solution for increasing the probability to obtain a global maximum instead of a local one. The usage of dynamically varying step sizes is also not new: it has been already reported in [Yuret and de la Maza, 1993; Miller, 2000]. Basically, any local search algorithm could have been used. RRHC, however, is because of its simplicity and straight forwardness the first choice for being adopted for optimizing several student model parameters. Other possible techniques are, for example, simulated annealing or genetic algorithms.

This work used the achievement prediction accuracy to measure the performance of student models. This is a common approach. Corbett and Anderson, for example, used the prediction accuracy to evaluate their knowledge tracing student model used in the ACT Programming Tutor [Corbett and Anderson, 1995]. Their performance prediction after every step depends on (1) the probability that a rule is in the learned state, (2) a slip parameter and (3) a guess parameter for the according rule. Desmarais and Gagnon compared the prediction accuracy of two cognitive student models: one based on a standard Bayesian network approach and the other on a more constrained one [Desmarais and Gagnon, 2006]. Previous versions of student models of ACTIVEMATH were evaluated by the prediction accuracy as well [Faulhaber, 2007; Faulhaber and Melis, 2008].

6 Conclusions and Future Work

This paper presented how the parameters of ACTIVEMATH's student model had been optimized with the local search algorithm known as random-restart hill climbing. Several parameters were presented, which were then optimized by a modified version of the RRHC algorithm.

[4]The reason why the prediction accuracy on the ALoE data is on average about 10% lower compared to the ATuF data is that the metadata quality of the ALoE data is worse and the number of difficult exercises is higher.

The modified RRHC uses dynamic step sizes and steadily switches between different parameters. The evaluation function replays log data to measure the achievement prediction accuracy of a configuration. The achievement prediction accuracy improved significantly by 3% for the ATuF data and by 2% for the ALoE data.

Until this work, the parameters of ACTIVEMATH's student model remained to be a manually chosen configuration. The evaluated parameter configuration did indeed improve SLM's accuracy. Since the performance of the original SLM was already quite good, an increase by 2% and 3% is an acceptable improvement. Nevertheless, improvements by 5% and above would have been more convincing.

The presented methodology can be generalized to optimize other user models as well. First of all, the set of parameters which have to be optimized has to be fixed. Additionally, an evaluation method has to be defined — replaying log data and measuring the prediction accuracy is just one possibility. Finally, RRHC can be used to find an optimized parameter configuration.

In the future we plan to use log data obtained from other projects using different exercises with participants of the same age as well as more experienced participants. This might help us to find out whether the obtained configuration is specific to the young learners or to the used exercises domain (which was fraction mathematics in both experiments, ATuF and ALoE). We want to further analyze the necessity of propagation and possible differences between age groups.

Acknowledgements

This article results from the ATuF project (ME 1136/5-2) funded by the German National Science Foundation (DFG).

References

[Aarts and Lenstra, 2003] Emile L. Aarts and Jan K. Lenstra. *Local search in combinatorial optimization*. Princeton Univ Pr, 2003.

[Cen *et al.*, 2007] Hao Cen, Kenneth R Koedinger, and Brian Junker. Is over practice necessary? –improving learning efficiency with the cognitive tutor through educational data mining. In *Proceeding of the 2007 conference on Artificial Intelligence in Education*, pages 511–518, Amsterdam, The Netherlands, 2007. IOS Press.

[Corbett and Anderson, 1995] Albert T Corbett and John R Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, 4:253–278, 1995.

[Desmarais and Gagnon, 2006] Michel C Desmarais and Michel Gagnon. *Bayesian Student Models Based on Item to Item Knowledge Structures*, pages 111–124. Springer-Verlag Berlin Heidelberg, 2006.

[Doost, 2010] Ahmad Salim Doost. Enhancement of activemath's student model. Master's thesis, Saarland University, 2010.

[Eichelmann *et al.*, 2008] Anja Eichelmann, Susanne Narciss, Arndt Faulhaber, and Erica Melis. Analyzing computer-based fraction tasks on the basis of a two-dimensional view of mathematics competences. In *EARLI SIG 6&7*, pages 125–134. Springer, 2008.

[Faulhaber and Melis, 2008] Arndt Faulhaber and Erica Melis. An efficient student model based on student performance and metadata. In Nikos Avouris, Nikos Fakotatis, Constantine D. Spyropoulos, and Malik Ghallab, editors, *Proceedings of 18th European Conference on Artificial Intelligence. 18th European Conference on Artificial Intelligence (ECAI-08), July 21-25, Patras, Greece*, volume 178 of *Frontiers in Artificial Intelligence and Applications, FAIA*, pages 276–280. University of Patras, IOS Press, 2008.

[Faulhaber, 2007] Arndt Faulhaber. Building a new learner model for activemath combining transferable belief model and item response theory. Master's thesis, Saarland University, 2007.

[Flynt and Morton, 2009] Samuel W Flynt and Rhonda Collins Morton. The teacher shortage in america: Pressing concerns. *National Forum of Teacher Education Journal*, 19(3), 2009.

[Gong *et al.*, 2010] Yue Gong, Joseph E Beck, and Neil T Heffernan. *Comparing Knowledge Tracing and Performance Factor Analysis by Using Multiple Model Fitting*. 2010.

[Ingersoll and Perda, 2009] Richard M Ingersoll and David Perda. The mathematics and science teacher shortage: Fact and myth. Technical Report CPRE Research Report #RR-62, The Consortium for Policy Research in Education, March 2009.

[Kohlhase, 2006] Michael Kohlhase. *OMDoc – An Open Markup Format for Mathematical Documents [version 1.2]: Foreword by Alan Bundy*. Lecture Notes in Computer Science. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[Lord, 1980] Frederic M. Lord. Applications of item response theory to practical testing problems. Hillsdale NJ: Erlbaum, 1980.

[Miller, 2000] Ronald E Miller. *Optimization: Foundations and Applications*. John Wiley & Sons, 2000.

[Russel and Norvig, 2003] Stuart Russel and Peter Norvig. *Artificial Intelligence: A Modern Approach*, pages 110–114. Prentice Hall, 2 edition, 2003.

[Tsovaltzi *et al.*, 2009] Dimitra Tsovaltzi, Erica Melis, Bruce McLaren, Michael Dietrich, George Goguadze, and Ann-Kristin Meyer. Erroneous examples: A preliminary investigation into learning benefits. In Ulrike Cress Vania Dimitrova Marcus Specht, editor, *Proc. of the First European Conference on Technology Enhanced Learning (EC-TEL 2009)*, volume LNCS 5794 of *Lecture Notes in Computer Science*, Heidelberg, 2009. Springer-Verlag.

[Ullrich and Libbrecht, 2008] Carsten Ullrich and Paul Libbrecht. *Educational Services in the ActiveMath Learning Environment*, pages 211–236. The Future of Learning. IOS Press Amsterdam, March 2008.

[Yuret and de la Maza, 1993] Deniz Yuret and Michael de la Maza. Dynamic hill climbing: Overcoming the limitations of optimization techniques. In *In The Second Turkish Symposium on Artificial Intelligence and Neural Networks*, pages 208–212, 1993.

On the Role of Social Tags in Filtering Interesting Resources from Folksonomies

Daniela Godoy
ISISTAN Research Institute, UNICEN University
Campus Universitario, CP 7000, Tandil, Argentina
Also at CONICET, Argentina
dgodoy@conicet.gov.ar

Abstract

Social tagging systems allow users to easily create, organize and share collections of resources (e.g. Web pages, research papers, photos, etc.) in a collaborative fashion. The rise in popularity of these systems in recent years go along with an rapid increase in the amount of data contained in their underlying folksonomies, thereby hindering the user task of discovering interesting resources. In this paper the problem of filtering resources from social tagging systems according to individual user interests using purely tagging data is studied. One-class classification is evaluated as a means to learn how to identify relevant information based on positive examples exclusively, since it is assumed that users expressed their interest in resources by annotating them while there is not an straightforward method to collect non-interesting information. The results of using social tags for personal classification are compared with those achieved with traditional information sources about the user interests such as the textual content of Web documents. Finding interesting resources based on social tags is an important benefit of exploiting the collective knowledge generated by tagging activities. Experimental evaluation showed that tag-based classification outperformed classifiers learned using the full-text of documents as well as other content-related sources.

1 Introduction

Social tagging systems have grown in popularity on the Web in the last years on account of their simplicity to categorize and retrieve shared content using open-ended tags. In sites such as *Del.icio.us*[1], *Flickr*[2] or *CiteULike*[3], users annotate a variety of resources (Web pages, blog posts or pictures) using a freely chosen set of keywords, which facilitates later search and retrieval of such contents.

Folksonomies [Mathes, 2004] are the primary structure of the novel social classification scheme introduced by tagging systems, which relies on the convergence of the tagging efforts of a large community of users to a common categorization system that can be effectively used to organize and navigate a massive amount of freely accessible, user contributed and annotated Web resources.

In spite of the novel mechanisms for searching and retrieving resources provided by collaborative tagging systems, the rapid increase in size of communities using these systems as well as the large amount of shared content available make the discovery of relevant resources a time consuming and difficult task for users. This problem is aggravated by the completely unsupervised nature of social tags, resulting in ambiguity, noise, etc.; which may reduce their effectivity in content indexing and searching.

The goal of this paper is to study the utility of social tags as a source of information for filtering resources from folksonomies according to the user interests. In social tagging systems resources receive tag assignments by members of the community, describing their content in a collective sense. Thus, it can be assumed that users are likely to be interested in additional content annotated with similar tags to the ones collectively assigned to resources they showed interest in before.

Social tags associated to the resources annotated by the user can be used to build a user interest profile that, in turn, can be applied to filter further incoming information from tagging systems (e.g. RSS feeds). From a user perspective, social tags can be thought of as indicators of user awareness and potential interest in a given resource [Arakji *et al.*, 2009], allowing users to capitalize on the associations made by persons who have assigned similar tags to other resources.

In order to identify interesting resources, tag-based classifiers are learned using the resources users annotate and have in their personomies, the tag collection of a single user, as positive examples of their interests. This is a special case of classification in which it is necessary to determine whether an example (resource) belongs to a target class (*interesting*) when only examples of the target class are given, which is known as one-class classification.

The rest of the paper is organized as follows. Section 2 gives an overview of one-class classification using Support Vector Machines (SVM) classifiers. Section 3 describes the dataset used for experimentation, gathered from *Del.icio.us* bookmarking site. The empirical analysis carried out to compare content-based and tag-based classification of Web pages in personomies is presented in Sections 4 and 5, respectively. Section 6 reviews related research. Finally, empirical findings are summarized in Section 7.

2 One-class Classification

User actions of assigning tags to resources are a strong indication of relevance about its content. Consequently, positive examples of the user interests can be easily collected from folksonomies. On the contrary, it would be

[1] http://del.icio.us/
[2] http://www.flickr.com/
[3] http://www.citeulike.org/

hard to identify representative negative examples or non-interesting resources since users might not tag a potentially interesting resource because of multiple reasons, such as not knowing about the existence of the resource, lack of time to tagging or even reading it, etc.

The task of determining whether a document is interesting for a user basing training only on positive examples can be seen as a one-class classification problem. One-class classification differs in one essential aspect from conventional classification as it assumes that only information of one of the classes, the target class, is available. The idea is to define a boundary between the two classes estimated from data belonging to the relevant class, such that it accepts as much of the target objects as possible while minimizes the chance of accepting outlier objects.

SVMs (Support Vector Machines) are a useful technique for data classification, which has been shown that is perhaps the most accurate algorithm for text classification, it is also widely used in Web page classification. Schölkopf et al. [Schölkopf *et al.*, 2001] extended the SVM methodology to handle training using only positive information and Manevitz et al. [Manevitz and Yousef, 2002] apply this method to document classification and compare it with other one-class methods.

Essentially, one-class SVM algorithm consists in learning the minimum volume contour that encloses most of the data and it was proposed for estimating the support of a high-dimensional distribution [Schölkopf *et al.*, 2001], given a set of training vectors $\mathcal{X} = \{x_1, \ldots, x_l\}$ in $\mathbb{R}^n$. The aim of SVM is to train a function $f_{\mathcal{X}} : \mathbb{R}^n \rightarrow \mathbb{R}$ such that most of the data in $\mathcal{X}$ belong to the set $\mathcal{R}_{\mathcal{X}} = \{x \in \mathbb{R}^n \text{ with } f_{\mathcal{X}}(x) \geq 0\}$ while the volume of $\mathcal{R}_{\mathcal{X}}$ is minimal. This problem is termed minimum volume set (MVS) estimation, and the membership of x to $\mathcal{R}_{\mathcal{X}}$ indicates whether this data point is overall similar to $\mathcal{X}$.

One-class SVM solves MVS estimation by first mapping the data into a feature space $\mathcal{H}$ using an appropriate kernel function $\phi : \mathbb{R}^n \rightarrow \mathcal{H}$ which transforms training examples to another space. Here, the Gaussian RBF kernel is used, formulated as $\exp\left[-\gamma \left\|x_i - x_j\right\|^2\right]$, where $\|.\|$ denotes the Euclidean norm in $\mathbb{R}^n$. For training, a certain number of data points of the positive class are treated as if they belong to the negative class. SVM approach proceeds in $\mathcal{H}$ by determining the hyperplane $\mathcal{W}$ that separates most of the data from the hypersphere origin, separating a certain percentage of outliers from the rest of the data points.

In order to separate the data points from the origin, the following quadratic programming problem needs to be solved:

$$\min_{w,\xi,\rho} \frac{1}{2}\boldsymbol{w}^T\mathbf{w} - \rho + \frac{1}{\nu l}\sum_{i=1}^{l}\xi_i$$

subject to

$$\boldsymbol{w}^T\phi(x_i) \geq \rho - \xi_i$$

$$\text{and } \xi_i \geq 0,\ i = 1, 2, \ldots, l$$

where ξ_i are so-called slack variables and υ (Nu) tunes the fraction of data that are allowed to be on the wrong side of $\mathcal{W}$, this parameter defines the trade-off between the percentage of data points treated as belonging to the positive and negative classes. Then a solution is such that α_i verify the dual optimization problem:

$$\min_{\alpha} \frac{1}{2}\boldsymbol{\alpha}^T Q\boldsymbol{\alpha} \tag{1}$$

subject to

$$0 \leq \alpha_i \leq 1/(\upsilon l), i = 1, \ldots, l$$

$$\boldsymbol{e}^T\boldsymbol{\alpha} = 1$$

where $Q_{ij} = K(x_i, x_j) \equiv \phi(x_i)^T \phi(x_j)$.

In this work we used LibSVM[4] [Chang and Lin, 2001] library which solves a scaled version of 2 as follows:

$$\min \frac{1}{2}\boldsymbol{\alpha}^T Q\boldsymbol{\alpha} \tag{2}$$

subject to

$$0 \leq \alpha_i \leq 1, i = 1, \ldots, l$$

$$\boldsymbol{e}^T\boldsymbol{\alpha} = \nu l$$

Finally, the decision function is:

$$\text{sgn}\left(\sum_{i=1}^{l}\alpha_i K(x_i, x) - \rho\right)$$

In order to adjust the kernel for optimal results, the parameter γ need to be tuned to control the smoothness of the boundary, i.e. large values of γ lead to flat decision boundaries. The setting of this parameter is initially set to $\gamma = 0$, variations of this value are then discussed in Section 5.

3 Dataset Description

Emerging social structures in tagging systems, also known as folksonomies, can be defined as a tuple $\mathbb{F} := (U, T, R, Y, \prec)$ which describes the users U, resources R, and tags T, and the user-based assignment of tags to resources by a ternary relation between them, i.e. $Y \subseteq U \times T \times R$ [Hotho *et al.*, 2006]. The collection of all tag assignments of a single user constitute a personomy, i.e. the personomy $\mathbb{P}_u$ of a given user $u \in U$ is the restriction of $\mathbb{F}$ to u, i. e., $\mathbb{P}_u := (T_u, R_u, I_u, \prec_u)$ with $I_u := \{(t, r) \in T \times R \,|\, (u, t, r) \in Y\}$, $T_u := \pi_1(I_u)$, $R_u := \pi_2(I_u)$, and $\prec_u := \{(t_1, t_2) \in T \times T \,|\, (u, t_1, t_2) \in \prec\}$, where π_1 the projection on the *i*th dimension.

Empirical evaluation was carried out using data collected from *Del.icio.us*[5] social bookmarking system. From this site 50 complete personomies were gathered from different users appearing in the main page. Each personomy includes all of the user bookmarks and the corresponding tag assignments. In this collection of personomies there are users with as few as 10 and as much as 2521 bookmarks. For each Web page, in turn, all tags assigned by other members of the community were also extracted from *Del.icio.us*, obtaining the full tagging activity (FTA) or annotations related to each resource.

From the total set of resources gathered from *Del.icio.us* site, experiments reported in this paper were performed over English-written pages, identified using the classification approach presented in [Cavnar and Trenkle, 1994]. This allows to apply language-dependent pre-processing

[4] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

[5] http://del.icio.us/

tasks to both texts and social tags. The resulting folksonomy counts with $|U| = 50$ users, $|T| = 233.997$ tags and $|R| = 49.265$ bookmarks or Web pages, related by a total of $|Y| = 128.642.112$ tag assignments. Table 1 summarizes the main statistics of this collection of Web pages averaged by personomy. It includes the number of unique terms in the full-text of resources belonging to the different personomies as well as in the text of anchors and titles. Also contains the number of tags assigned by members of the community to the resources of each user, considering the overall top 10 tags and the full tagging activity. The effect produced in these numbers by two tag filters explained in Section 5 is also detailed. The average numbers of each element correspond to the number of features classifiers have to deal with during learning.

In all experiments reported in this paper, evaluation was carried out using a holdout strategy that split data into a 66% for training and a 34% for testing. In order to make the results less dependent of the data splitting, in all experiments the average and standard deviation of 10 runs for each user is reported. This is, each personomy was divided into a training set used to learn the classifier and a testing set used to assess its validity. Since this testing set only contains interesting examples, uninteresting pages were extracted from the personomies of other users to evaluate the algorithm capacity of distinguishing uninteresting resources. This is, the testing set was created using the test set from the user and an equivalent number of Web pages gathered from a different personomy in the collection. This second personomy was randomly chosen among those presenting no resource intersection with the current user. In other words, it is assumed that two users having no common resources in their personomies do not shared interests, so that one user resources will be uninteresting to the other one. Although this is not strictly true, it can be considered as an approximation to obtain a negative set for testing. For evaluating the classifiers, the standard precision and recall summarized by F-measure as well as accuracy were used and error-bars indicate standard deviations [Baeza-Yates and Ribeiro-Neto, 1999].

4 Content-based Classification

Content is one of the main sources of information for determining the relevance of Web pages for users. It is assumed that similar contents to those previously seen by the user will be also interesting. In order to establish the relative importance of content and social tags in personal Web page classification, the performance of one-class classification over textual elements obtained from documents was first evaluated so that it can be used as baseline for comparing the performance of tag-based classifiers.

Web page texts were filtered using a standard stop-word list and the Porter stemming algorithm [Porter, 1980] was applied to the remaining terms. Figure 1 shows the results of training classifiers for identifying interesting Web pages using different textual sources such as the full text of documents, the anchor text attached to hyperlinks (i.e. the visible, clickable text in a hyperlink) belonging to the page and the page title. Each of these elements is extracted from pages belonging to a user personomy to learn a classifier for such user. F-measure scores achieved with different values of ν (Nu) parameter of one-class classifiers are showed in the figure.

Classification using full-text obtained the best results, closely followed by the text from anchors. The title of resources alone, however, did not result to be a good source for filtering interesting information. Naturally, the relatively low scores of F-measure is caused by the absence of negative information during learning. In addition, in the negative testing set might be some interesting pages, due to possible violations of the assumption that users do not shared interests if their personomies do not intersect each other, for which the prediction is correct but taken as an error. Nevertheless, text classifiers were still able of recognizing part of the user interests and are a valuable source for filtering a stream of incoming information (e.g. a RSS feed).

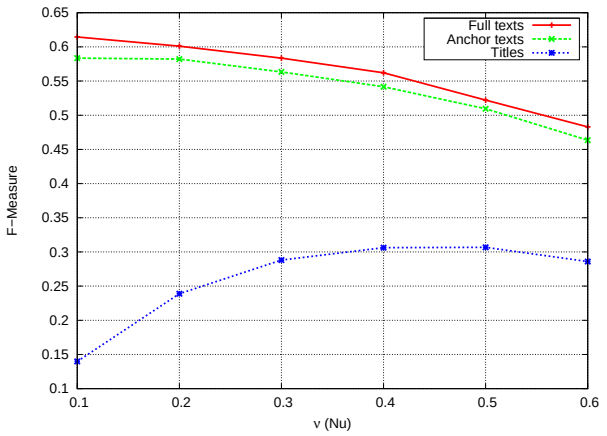


Figure 1: F-measure scores using different textual sources from the content of resources for classification

5 Social-based Classification

Social tagging systems on the Web own their success to the opportunity of freely determining a set of tags for a resource without the constraint of a controlled vocabulary, lexicon or pre-defined hierarchy [Mathes, 2004]. However, the free-form nature of tagging also leads to a number of vocabulary problems. Among the reasons producing tags variations are [Golder and Huberman, 2006; Tonkin and Guy, 2006; Echarte *et al.*, 2008]:

- inconsistently grouping of compound words consisting of more than two words. Often users insert punctuation to separate the words, for example *ancient-egypt*, *ancient_egypt* and *ancientgypt*;
- use of symbols in tags, symbols such as #, -, +, /, :, _, &, ! are frequently used at the beginning of tags to cause some incidental effect such as forcing the interface to list some tag at the top of an alphabetical listing;
- morphological problems given by the use of singular, plural or other derived forms of words. For example, *blog*, *blogs* and *blogging.*

To prevent syntactic mismatches due to these reasons the effect of different filtering strategies for tags was evaluated. First, original raw tags were filtered to remove the symbols mentioned before, allowing to join compound words at the same time. Then, the remaining tags were stemmed to their morphological roots using Porter stemming algorithm.

In this study the overall top 10 tags associated to resources in the folksonomy, this is the 10 more frequent tags per resource, were evaluated as a source for classification and compared with the use of the complete set of tags assigned for users to such resources, also known as the full

	Min	Max	Average	±	SD	Total
# full-text terms	1.997	115.585	47.138,14	±	29.063,53	2.356.907
# anchor text terms	681	62.521	24.632,82	±	16.268,12	1.231.641
# title terms	31	4.581	1.806,44	±	1.236,02	90.322
# tags in the top 10 lists	57	4.117	1.739,76	±	1.097,52	86.988
# tags in the top 10 lists after filtering symbols	57	3.882	1.686,14	±	1.055,23	84.307
# tags in the top 10 lists after stemming	55	3.462	1.495,68	±	934,83	74.784
# tags in the FTA	150	10.902	4.679,94	±	3.053,78	233.997
# tags in the FTA after filtering symbols	141	9.872	4.328,46	±	2.791,12	216.423
# tags in the FTA after stemming	122	8.678	3.757,00	±	2.426,63	187.856

Table 1: Summary of Web page statistics per personomy in the dataset used for experimentation

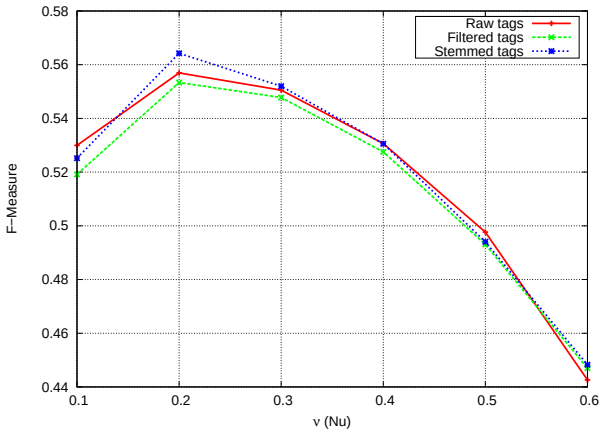


Figure 2: F-measure scores using frequency-based representations of the top 10 tags associated to resources for classification

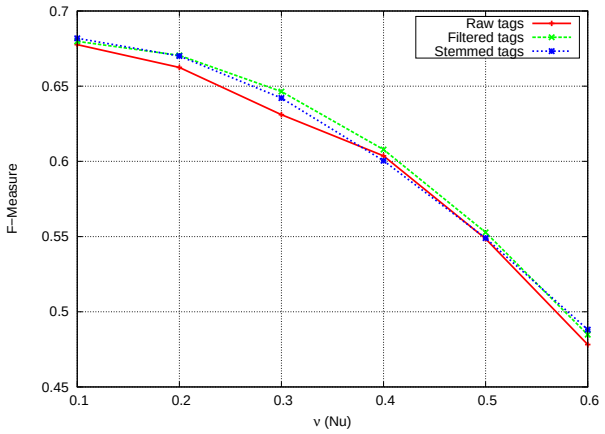


Figure 3: F-measure scores using binary representations of the top 10 tags associated to resources for classification

tagging activity of the resource. Frequency-based and binary representations of the resulting tag vectors were also considered and compared. Binary vectors were constructed to indicate the occurrence or non-occurrence of a given tag in the list of tags a Web page is annotated with. Frequency vectors indicate the number of users that employ a given tag to annotate the resource; this is f_{ij} is the frequency of usage of tag *i* for the resource *j,* these vectors are normalized according to their length.

5.1 Results using top-10 tags

Figure 2 depicts F-measure scores achieved with one-class SVM classifiers leaned using the top 10 list of tags. In the figure, results are shown for raw tags as well as tags resulting of applying the mentioned filtering strategies, first symbol removal and then stemming. In regards to the tag filtering operations it can be deduced according to these results that removing symbols and joining compound words slightly diminish the performance of classifiers, whereas stemming improves it. Note that the filters were applied in the previously mentioned order, so that stemming can potentially achieve better results if applied over raw tags directly.

In general terms, the results of using binary representation for tag vectors, which are shown in Figure 3, provides a significant improvement over normalized frequency vectors. In this representation scheme, removing symbols and joining compound words reduce the noise of tags resulting in an improvement of F-measure scores. However, the use of stemming does not lead to further improvements, even damaging the performance of classifiers.

5.2 Results using full tagging activity

Figures 4 and 5 depict the results using the same configuration of experiments but applied to vectors resulting of the full tagging activity attached to resources.

F-measure scores of one-class SVM classifiers learned using frequency vectors, depicted in Figure 4, shows that the filter used to remove symbols and join compound words does not improve significantly the performance of classifiers whereas stemming obtains slightly enhancements. In turn, binary representations again outperformed frequency-based ones and filters attain small performance enhancements over raw tags.

5.3 Summary of results

Figure 6 summarizes the results obtained for full-text classification of Web resources and tag-based classification using both the top 10 tags of each resource and its full tagging activity in their frequency-based and binary representations.

It can be observed that the results of using the main source of information about the content of a resource, which is the text of the resource itself, is consistently outperformed by the use of social tags when a binary representation of tag vectors is applied. Classification based on frequency-based representations of the full tagging activity of resources also reached better performance than full-text classification. In contrast, the use of the 10 more frequent tags used to annotate resources as a means for classification exhibit inferior performance in identifying interesting Web pages for users.

Figure 7 shows the results obtained with the same classification sources by setting ν to 0.1, the point at which

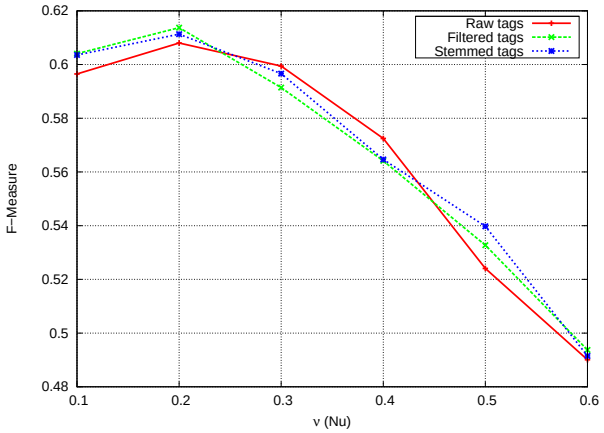


Figure 4: F-measure scores using frequency-based representations of the full tagging activity for classification

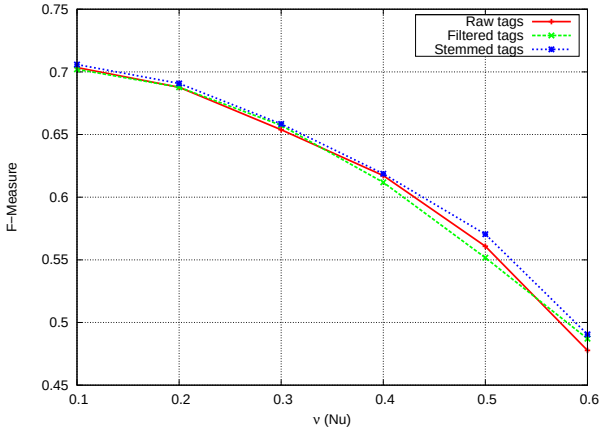


Figure 5: F-measure scores using binary representations of the full tagging activity for classification

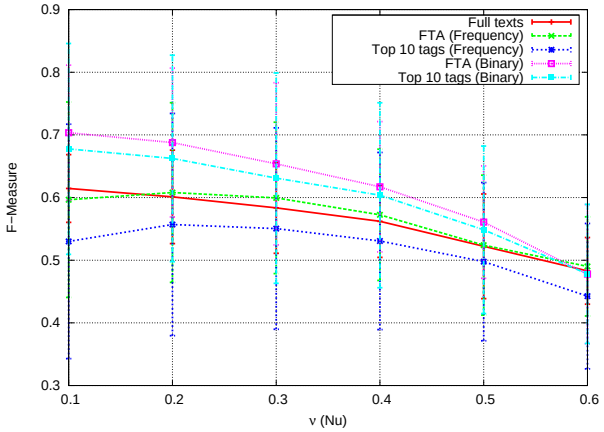


Figure 6: F-measure scores considering content and social tags as sources for classification

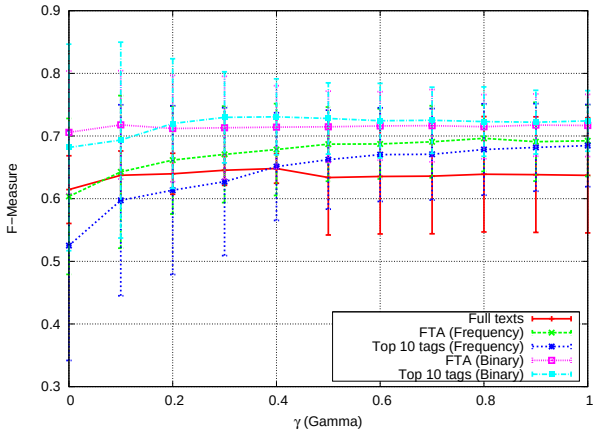


Figure 7: F-measure scores for variations of γ (gamma) parameter of one-class SVM classifiers

the best results were achieved, and varying the value of γ of one-class SVM classifiers. The figure not only shows how higher values of γ lead to small increases in F-measure scores but, more importantly, with values of $\gamma > 0.4$ any form of representation of social tags outperforms full-text classification. Furthermore, binary representations of top-10 tags associated to resources become the best performing among the social classification schemes.

It is worth mentioning that full-text is used in these experiments as baseline for comparison, but this source of information is not always available in social tagging systems in which resources can be a variety of things, such as images, music, bibliographic references, etc. In these situations, classification must entirely rely on social tags. Thus, it can be concluded that collective knowledge lying in folksonomies becomes a valuable source of information for automatic, personal classification of Web resources.

Learning classifiers using collaboratively assigned tags also impacts on the dimensionality of the classification problem. In fact, tag-based classifiers extracted from the top 10 list of tags are learned in a smaller dimensional space than full-text classifiers and yet are better predictors as can be observed in the last results reported. Table 1 summarized the number of unique features, terms or tags according to the case, the classification problem have to deal with.

Figure 8 summarizes the performance of content and tags-based classifiers in terms of accuracy for $\upsilon = 0.1$, the parameter setting leading to the best results in most experiments. Confirming previous results, if the classifiers capability of making correct decisions is considered, tags-based classifiers outperformed full-text ones. Also, among tag-based classifiers those using for training the top 10 tags assigned to each resource were the ones of superior performance. Thus, top 10 tags offers good accuracy levels and, at the same time, an important reduction in leaning and prediction complexity given the smaller size of the dimensional space.

Finally, the incidence of the different sources of information for filtering Web pages, content or social tags, is analyzed according to the size of personomies in Figure 9. For studying this aspect of classification, the 50 users were divided in five groups according to the amount of resources in their personomies. In the first group users having less than 300 annotated resources were placed, then users having from 300 to 600 resources, 600 to 1000, 1000 to 2000, and more than 2000 resources.

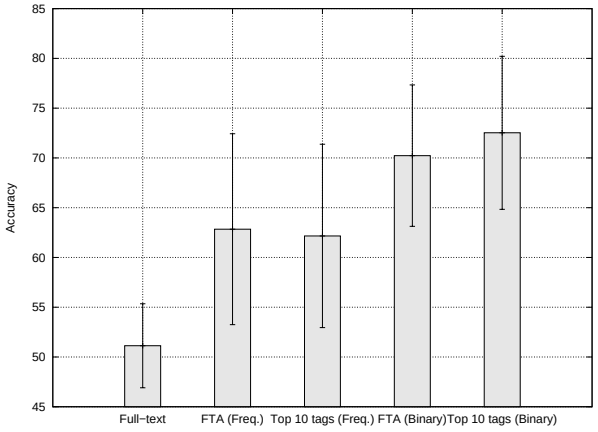


Figure 8: Accuracy of content and social tags classifiers

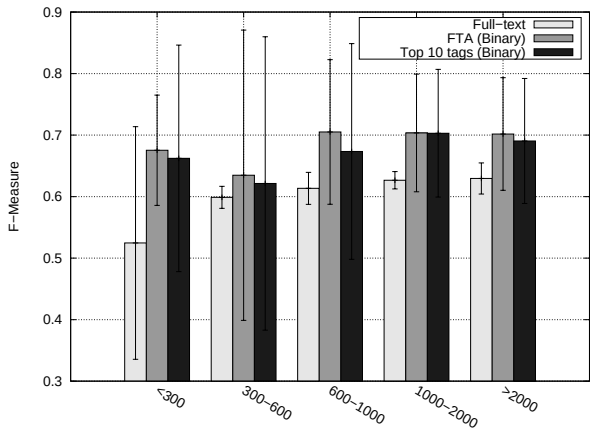


Figure 9: F-measure scores obtained for different personomy sizes

Naturally, as the number of annotated resources grows classifiers becomes increasingly better in filtering interesting Web pages as more information about the user interests is available during learning. However, the difference between social tags and full-text classifiers is more noticeable in smaller personomies, in which tags outperform text by a wider margin.

6 Related Works

Many works had approached the problem of tag recommendation in social tagging systems [Lipczak, 2008; Symeonidis *et al.*, 2008; Jäschke *et al.*, 2007; Milicevic *et al.*, 2010], however the problem of filtering resources according to the user interest had received less attention. To the best of our knowledge, no approaches have addressed this problem using one-class classification over social tags.

Vatturi et al. [Vatturi *et al.*, 2008] create a personalized ed tag-based recommender for each user consisting of two NB classifiers trained over different time-frame. One classifier predicts the user current interest based on a shorter time interval and the other classifier predicts the user general interest in a bookmark considering a longer time interval. If any classifier predicts the bookmark as interesting, it is recommended. The user study results show that the tag-based recommender performs well with real data using tags from an enterprise social bookmarking system. In [Ammari and Zharkova, 2009] an approach for filtering the blog posts that search engines retrieve is presented. SVM is used to train and build a predictive model for the targeted user; the retrieved posts are analyzed and classified by the predictive model. Finally, only the posts that are scored as relevant by the model are sent back to the user.

Tag-based profiling consisting on tag vectors in which tag weights are given by their frequency of occurrence in the resources a user tagged had been proposed in [Noll and Meinel, 2007]. In [Michlmayr and Cayzer, 2007], profiles are represented by graphs in which nodes correspond to tags and edges denote relationships between them. The idea of using semantic relationships among tags in tag-based profiles has also been explored in [Huang *et al.*, 2008]. In the work presented in this paper, one-class SVM classifiers can be seen as tag-based profiles for users.

The value of the collective knowledge encapsulated in social tags for classification of resources in general directories or categories was studied in several works, not from a personal perspective as in this work, but from a social point of view.

Zubiaga et al. [Zubiaga *et al.*, 2009] explore the use of Support Vector Machines (SVM) in the *Social-ODP-2k9* dataset, which links Web pages and tags assigned to them in *Del.icio.us* with their corresponding categories in a Web directory such as the *Open Directory Project* (ODP)[6]. In this work additional resource meta-data such as notes and reviews were also evaluated besides the tagging activity. Tags in conjunction with comments achieved good results for Web page classification.

Noll and Meinel [Noll and Meinel, 2008b] study and compare three different annotations provided by readers of Web documents, such as social annotations, hyperlink anchor texts and search queries of users trying to find Web pages, for classification. Coincidentally with our finding in the context of personal Web page classification, the results of this study suggest that tags seem to be better suited for classification of Web documents than anchor words or search keywords, whereas the last ones are more useful for information retrieval. In a further study [Noll and Meinel, 2008a], the same authors analyzed at which hierarchy depth tag-based classifiers can predict a category using the ODP directory. It was concluded that tags may perform better for broad categorization of documents rather than for narrow categorization. Thus, classification of pages in categories at inferior hierarchical levels might require content analysis.

7 Conclusions

In this paper the role of social tags in filtering resources from folksonomies according to the interests of individual users was empirically analyzed. One-class classification was used to learn the user interests from diverse content sources (such as the full-text, anchor texts and titles) and social tagging sources (top 10 list of all tags associated to resources and their full tagging activity). Then, the extend to which each source can contribute to automatic, personal Web document classification was evaluated and compared.

Experimental results obtained with a set of personomies extracted from *Del.icio.us* bookmarking system showed that tag-based classifiers outperformed content-based ones. Some tag filters such as removal of symbols, joint of compound words and reduction of morphological variants have

[6] http://www.dmoz.org/

a discrete impact on classification performance. Interesting results were obtained using binary representations of tag vectors for learning and prediction. In this case, tag-based classifiers significantly improved the performance in filtering interesting results, even considering the top 10 tags assigned to resources in a quite smaller dimensionality space.

Acknowledgments

This research was supported by The National Council of Scientific and Technological Research (CONICET) under grant PIP Nº 114-200901-00381.

References

[Ammari and Zharkova, 2009] A. N. Ammari and V. V. Zharkova. Combining tag cloud learning with SVM classification to achieve intelligent search for relevant blog articles. In *Proceedings of the 1st International Workshop on Mining Social Media*, Sevilla, Spain, 2009.

[Arakji *et al.*, 2009] R. Arakji, R. Benbunan-Fich, and M. Koufaris. Exploring contributions of public resources in social bookmarking systems. *Decision Support Systems*, 47(3):245–253, 2009.

[Baeza-Yates and Ribeiro-Neto, 1999] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley Longman Publishing, 1999.

[Cavnar and Trenkle, 1994] W. Cavnar and J. Trenkle. N-gram-based text categorization. In *Proceedings of 3rd Annual Symposium on Document Analysis and Information Retrieval*, pages 161–175, Las Vegas, USA, 1994.

[Chang and Lin, 2001] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[Echarte *et al.*, 2008] F. Echarte, J. Astrain, A. Córdoba, and J. Villadangos. Pattern matching techniques to identify syntactic variations of tags in folksonomies. In *Proceedings of the 1st World Summit on The Knowledge Society (WSKS '08)*, volume 5288 of *LNCS*, pages 557–564. Springer-Verlag, 2008.

[Golder and Huberman, 2006] S. Golder and B. Huberman. Usage patterns of collaborative tagging systems. *Journal of Information Science*, 32(2):198–208, 2006.

[Hotho *et al.*, 2006] A. Hotho, R. Jáschke, C. Schmitz, and G. Stumme. Information retrieval in folksonomies: Search and ranking. In *The Semantic Web: Research and Applications, 3rd European Semantic Web Conference, ESWC 2006*, volume 4011 of *LNCS*, pages 411–426. Springer, 2006.

[Huang *et al.*, 2008] Y-C. Huang, C-C. Hung, and J. Yung-Jen Hsu. You are what you tag. In *AAAI Spring Symposium on Social Information Processing (AAAI-SIP)*, pages 36–41, 2008.

[Jäschke *et al.*, 2007] R. Jäschke, L. Marinho, A. Hotho, L. Schmidt-Thieme, and G. Stumme. Tag recommendations in folksonomies. In *Knowledge Discovery in Databases: PKDD 2007*, volume 4702 of *LNCS*, pages 506–514, 2007.

[Lipczak, 2008] M. Lipczak. Tag recommendation for folksonomies oriented towards individual users. In *Proceedings of ECML PKDD Discovery Challenge (RSDC08)*, pages 84–95, Antwerp, Belgium, 2008.

[Manevitz and Yousef, 2002] L. M. Manevitz and M. Yousef. One-class SVMs for document classification. *Journal of Machine Learning Research*, 2:139–154, 2002.

[Mathes, 2004] A. Mathes. Folksonomies - cooperative classification and communication through shared metadata. Computer Mediated Communication, 2004.

[Michlmayr and Cayzer, 2007] E. Michlmayr and S. Cayzer. Learning user profiles from tagging data and leveraging them for personal(ized) information access. In *Proceedings of the Workshop on Tagging and Metadata for Social Information Organization*, Banff, Alberta, Canada, 2007.

[Milicevic *et al.*, 2010] A. K. Milicevic, A. Nanopoulos, and M. Ivanovic. Social tagging in recommender systems: A survey of the state-of-the-art and possible extensions. *Artificial Intelligence Review*, 33(3):187–209, 2010.

[Noll and Meinel, 2007] M. G. Noll and C. Meinel. Web search personalization via social bookmarking and tagging. In *Proceedings of 6th International Semantic Web Conference (ISWC) and 2nd Asian Semantic Web Conference (ASWC)*, volume 4825 of *LNCS*, pages 367–380, 2007.

[Noll and Meinel, 2008a] M. G. Noll and C. Meinel. Exploring social annotations for Web document classification. In *Proceedings of the 2008 ACM Symposium on Applied Computing (SAC '08)*, pages 2315–2320, Fortaleza, Ceará, Brazil, 2008.

[Noll and Meinel, 2008b] M. G. Noll and C. Meinel. The metadata triumvirate: Social annotations, anchor texts and search queries. In *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, pages 640–647, Sydney, Australia, 2008.

[Porter, 1980] M. Porter. An algorithm for suffix stripping program. *Program*, 14(3):130–137, 1980.

[Schölkopf *et al.*, 2001] B. Schölkopf, J. C. Platt, J. C. Shawe-Taylor, A. J. Smola, and R. C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7):1443–1471, 2001.

[Symeonidis *et al.*, 2008] P. Symeonidis, A. Nanopoulos, and Y. Manolopoulos. Tag recommendations based on tensor dimensionality reduction. In *Proceedings of the 2008 ACM Conference on Recommender Systems (RecSys '08)*, pages 43–50, Lausanne, Switzerland, 2008.

[Tonkin and Guy, 2006] E. Tonkin and M. Guy. Folksonomies: Tidying up tags? *D-Lib*, 12(1), 2006.

[Vatturi *et al.*, 2008] P. K. Vatturi, W. Geyer, C. Dugan, M. Muller, and B. Brownholtz. Tag-based filtering for personalized bookmark recommendations. In *Proceeding of the 17th ACM Conference on Information and Knowledge Management (CIKM '08)*, pages 1395–1396, Napa Valley, California, USA, 2008.

[Zubiaga *et al.*, 2009] A. Zubiaga, R. Martínez, and V. Fresn. Getting the most out of social annotations for Web page classification. In *Proceedings of the 9th ACM Symposium on Document Engineering (DocEng' 2009)*, pages 74–83, Munich, Germany, 2009.

User Models meet Digital Object Memories in the Internet of Things

Dominikus Heckmann
DFKI GmbH, Stuhlsatzenhausweg 3
66123 Saarbrücken, Germany
Dominikus.Heckmann@dfki.de

Abstract

In this paper, we argue that Digital Object Memories in the Internet of Things are closely related to partial user models in Personalization and that research can gain insights by analogy from both sides. We describe UbisMemory, a Semantic Web middleware for partial user models and digital object memories. We describe the content representation, the service and its technical issues. We argue that Digital Object Memories can be extended and merged with "Digital User Memories" or life-long user models. We argue that Life-Logging for objects and for humans are closer related than expected.

Introduction

"Personalization and Recommendation on the Web and Beyond" is the new title of the ABIS workshop series. In this paper we are looking at the word "Beyond" and what could be meant by it. We move from the Web to the real world and return back: the so called "Internet of Things". We look at how memories of users and memories of objects differ, interact and become blurred. User modeling aspects have recently turned more and more into a broader view of "context-awareness". This important focus on the context reveals that human-computer interaction takes place in different environments – with the immense increase of mobile technology, this context becomes more and more the real world itself. The desktop metaphor has been replaced by life-long user modeling in the real world.

A second path to this paper comes from the other direction. The concept of Digital Object Memories (DOM) has recently been introduced, see [1] or [7]. In the upcoming and fast growing research area "Internet of Things" (IoT), the "Digital Object Memories" play a central role to enable its ideas, strategies and goals. Embedded systems, mostly based on Semantic Web Technologies, Context-Awareness and intelligent identification mechanisms enable the technical implementation. Intelligent, instrumented environments are the foundation, where the digital object memories develop and grow. The dimension of storage or location of object memories is orthogonal to its options and possibilities that are enabled by the highly context aware environments. That means, at the service layer, it is not important if the digital object memory is stored directly locally with the object, or even if the identification of the object is defined directly locally with the object. Important is only that each object can be identified uniquely by the service layer and that the object's memory can be accessed somehow. In the internet everything that can be uniquely identified is called a resource. Thus we talk about memories attached to resources.

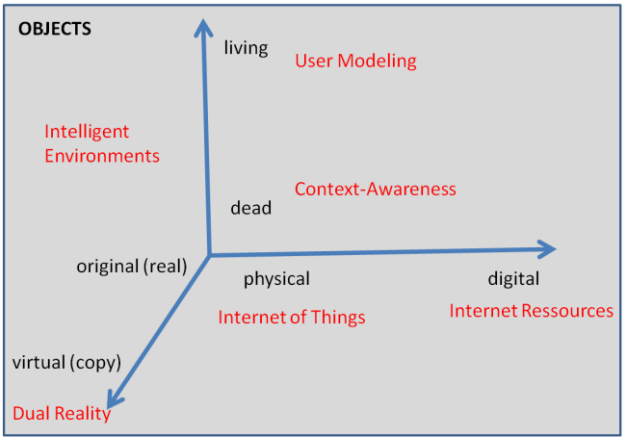


Figure 1: Object Dimensions for Digital Memories of Users and Objects, in relation to the reseach areas

If we abstract from technical issues like RFID, power supply, networking, sensors etc. at least two philosophical questions could arise between user models and digital object memories.

a) If we look at the persuasion in artificial intelligence of "Materialism" and "Strong AI", there is no difference concerning the potentiality of intelligence between a "dead" object and a "living" object like a human being. Thus, Digital (INANIMATE) Object Memories and Digital (LIVING) Object Memories can be considered as equivalent. The latter one has been researched since decades under the term of "user modeling".

b) A second philosophical issue might arise: do non-living objects get a personality as soon as they store their history in a DOM (of course with the help of the sensors and actuators in the intelligent environments)? To support the close relation between User Modeling and Digital Object Memories we could have also taken an approach from the opposite direction and make the "gedankenexperiment" and personify dead objects as soon as they manage a memory. The statement

by Tara Bloom "*Memory maketh the man* – we are who we are thanks to our experiences."

What matters for this paper and the UbisMemory service is the possibility of uniform handling of Digital Memories, for objects and humans. Thus the user models meet the digital object on the technical layer, however if one is able to adapt the materialism's point of view, also on the philosophical layer.

One advantage of this approach is that we can later introduce a uniform handling of privacy issues, like the so called onion model.

Important is the handling of "Digital Memories", independent, if they describe an object, a human, or an immaterial resource, like an entity on the web, or even a simulated resource in dual reality.

The developed middleware mainly looks at the management of "Digital Memories", independently if they are carried along, distributed in the environment or centrally stored

The viewpoint matters and DOMs have a kind of "object-oriented" viewpoint: nor the software-system is in the center, that handles all information and that communicates with all users, neither the intelligent environment. In the center is only the object, looking out to the world and managing its "personal" memory.

In Figure 1 we try to show that user modeling is closely related to the Internet of Things, if we manage to broaden the object dimensions.

Farm Scenario to look at DOM issues

Imagine you want to transform a farm into an intelligent environment and you want to apply the technology of digital object memories for a use case in the rural farming production of eggs. One question that arises is WHAT is the object? For which objects is it interesting to manage DOMs? Well it could be the hen, it could be the egg that will be eaten by the consumer. It could be the product, to say the box of eggs that will be bought by the consumer. It could be the chicken run. To say, it could actually be everything. It depends on the use case, the business case or the application. Looking through the glasses of "intelligent environments": everything in the context is an object of interest, to be on the save side. Looking through the glasses of "user modeling", especially the farmer is of interest. However, the farmer's user model could be based on all the other objects memories, if we take the context into account.

Interesting is the issue of knowledge representation and knowledge exchange in the Internet of Things, in comparison to the knowledge representation and knowledge exchange in Personalization. The following situation serves as basis to discuss the issue of object memories in this paper:
The hen "Lilli" has layed an egg yesterday in its barn in Elm, Germany. At that time is was 25 degrees Celsius. The Egg is packed with 5 others into a box, which is put on a pallet, which is moved by farmer Bob with a tractor from the farm „Erlenhof" to the merchants place.

Which parts of this information can be sensed by which sensors, which parts should be stored where? Should this information be distributed to the different objects or resources (hen, egg, box, pallet, farmer, farm …) that might collect data for their Digital Object Memories? Or should the description of the situation be stored in this intelligent farm environment. Or does it even make sense to introduce a centralized storage service and broker for partial digital object memories and partial user models?

The rest of this paper is arranged as follows: we discuss the issue of storage of object memories in the next section, and introduce parts of the UbisMemory system in section three, followed by a concluding discussion.

Storage of Object Memories

A first question that arises is where should the object's memory be stored? Well, an obvious solution is to attach it directly to the object itself. Interestingly, for the eggs in the farm scenario is that parts of the object memory are written on the eggs directly. One can read the date, the country, even the chicken run where the egg was laid (of course not in digital form).

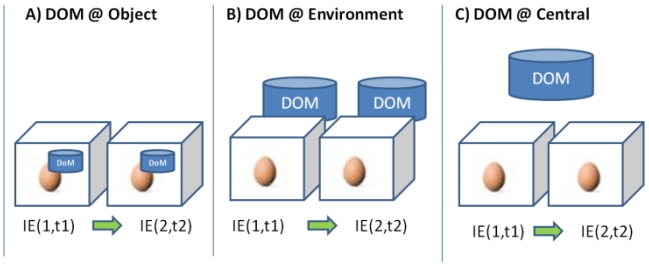


Figure 2: Local (A), Distributed (B), versus Central (C) Storage (IE = Intelligent Environment)

What could be the advantage of a central DOM repository? Well imagine the egg is further processed, like a scrambled egg. What to do if the consumer gets an egg-related disease after a couple of days? In that case It would be especially important to find out all the DOM-details like "the chicken run" that produced the egg.

In that scenario, the DOM gets interesting days after the object has been destroyed. If we only rely on A), where the DOM is stored directly at the object, we have a problem. An additional C) DOM @ Central would be of great help. Another argument of a centralized DOM-archive is the possibility of a uniform user interface to inspect the memories and to set privacy issues.

As conclusion, all three versions A), B) and C) have their rights of existence, their advantages and disadvantages. The UbisMemory service that is described hereafter focuses firstly on the centralized C) and secondly on distributed B).

UbisMemory

UbisMemory mainly addresses the issue of "Content Sharing" within ubiquitous computing, that can be compared to [1]. The UbisMemory middleware is integrated into the test bed architecture called UbisWorld 3.0, see [8], for research on ubiquitous user modeling in the era of

Semantic Web and Web 2.0 facing the Internet of Things in ubiquitous computing.
The basic storage idea is founded on **SituationalStatements** that form a relation-based user model & context representation, see[5]. The exchange protocol is defined as the earlier introduced **UserML,** which is an XML & RDF-based exchange language. Interesting is the question of what will be exchanged? Apart from the "Mainpart", also Meta Data are exchanged.
Which meta data is important for ubiquitous user modeling?

- When and where is the statement valid?
- Who claims this and which explanation is given?
- What is the evidence and the confidence?
- What will be done with the DOM/UM?
- When will this information be deleted?
- Who is the owner of this information?
- What are the privacy settings?
- How can the statement be uniquely identified?
- Can the DOM/UM entries be grouped with others?

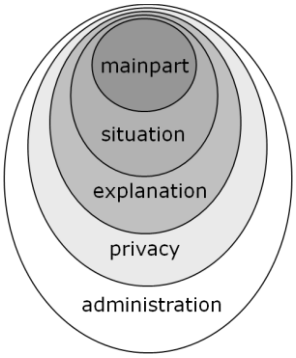


Figure 3: onion model of statements with several layers of meta-data

This meta-data can be used to define filters to tailor the parts of DOMs in the case of a DOM request.

Instead of describing the full UbisMemory architecture, see [8] for more details, we pick out the following three topics and describe them in more detail: privacy issues, exploitation and the user interface.

UbisMemory supports a fine grained semantic Location Modelling for DOMs. It is based on HUGO, the "Huge Ontology" with currently over 20 Million semantically described places. HUGO, see [7], is based on Linked Data, enriched with an Web 2.0 approach to add collaboratively new locations. It ranges from countries, cities, streets to individual houses, rooms and even shelves in furniture. Especially for detailed DOMs, a fine-grained location model is needed that goes beyond the GPS coordinates or the position with two dimensional coordinates, since interaction often takes place inside a house or a production place.

DOM Exploitation

To consider systems that integrate adaptation as exploitation of DOMs and the context is introduced in [9]. In [7] it is shown that complex applications can still be realized with cheap and weakly instrumented smart objects. This is done by establishing digital object memories.
How can the UbisMemory be integrated into the full process loop of enabling smarter and more intelligent applications on the basis of digital object memories? Well, in order to exploit the Digital Object Memories, a full process loop of the following four steps have to be implemented.

1) Instrumentation (i.e of the farm, or the farmer)
2) Interpretation (inference from sensor data to memory)
3) Communication (exchange of DOMs)
4) Adaptation (exploitation of DOMs & context)

Some techniques are discussed in [4] on how the interpretation from Sensor Data to Memories via LOGs and JOURNALS can be done.
UbisMemory only applies to the third step of "communication". In this sense it serves as a broker of DOMs. They can be stored there & retrieved from there. Together with a privacy handling mechanism, a broker service has been established.

Privacy Issues in UbisMemory

The exploitation of the resulting memories need to be regulated. In the farm scenario: *For example, who should be allowed to read the information on what fodder was given to the hen that layed the egg?*
We can support fine-grained privacy handling, since the SituatinalStatements apply the onion model which means that the mainpart information is only available if the privacy filter has been passed. A detailed description can be found in [6].

UbisMemory's User Interface

We have developed a general user interface to inspect the digital object memories as well as partial life-long user models, (as discussed in section one). Figure 4 shows a screenshot. The user interface can be accessed[1] via http://ubisworld.org/statementstore/.

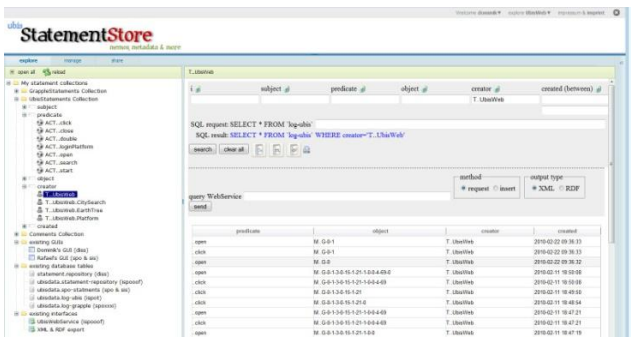


Figure 4: ubis.StatementStore, the user inteface of UbisMemory to inspect and control the Digital Memories of Objects and Users.

[1] however certain access rights need to be requested first.

The key idea is to decouple every situational statement into its semantic roles and allow for each of them a filtering by selection. By this approach you can select every object, and every attribute value pair that describes the memory item in mind, together with its meta-level information like timestamp, location, and privacy information like the owner of this bit of information.

CONCLUSION

In this paper we argue that user models will eventually meet digital object memories in the internet of things. In [2] we have stated that: *a comprehensive log of the user's behavior together with corresponding context descriptions allows adaptive systems to learn about users, to identify their habits, and to improve the quality of user support. In addition, users can apply such knowledge to learn from others and about themselves.* Now, how much more interesting and fruitful can it be if we log the objects behavior's, situations and contexts with the overall approach of Digital Object Memories become together with the Internet of Things? We expect a better quality of object support and a better quality of the objects (or products) themselves. For example, if we transfer the onion model of statements from user model research to digital object memories. Highly interesting are social implications and philosophical issues if we assume that objects gain a certain level of personality by their new memories. Instead of looking at Digital Object Memories only, we looked at "digital memories of objects", while we extend the view of "object" to the three dimensions physical-digital, real-virtual, and inanimate-living. With this generalization, we are able to combine User Modeling and Context-Awareness even in Dual Reality scenarios under the integrating concept of "Digital Memories of Objects". Thus, classical Digital Object Memories and Life-Long User Models can be treated in our approach with the same middleware.

In this paper we tried to contribute to the four following topics: **Memory Representation**: which we realize with SituationalStatements as introduced in [5]. **Memory Architectures**: We pointed to UbisMemory, a middleware approach which also allows for the realization of object memory functionality. This includes infrastructures for the centralized or distributed organizing, storing, and brokering of object-related information, however based on a remote infrastructure, not on the object itself. **Privacy Aspects**: Who "owns" the data stored in an object's memory, who can access/delete/correct it? How long must/should memory content be stored, and can trust be established for the object memory? **Human Memory Access:** This topic comprises technologies and concepts to make an object memory's content accessible to human users. With the UbisMemory browser, called StatementStore, we try to structure and relate the wide variety of diverse data that might be contained in the memory due to its open nature. The presented middleware architecture is integrated into an interesting test bed for research on ubiquitous user modeling in the era of Semantic Web and Web 2.0 facing the Internet of Things in ubiquitous computing. This UbisWorld 3.0 tool set can be tested online at www.ubisworld.org. What is "beyond" the Web? Well, the real world! However, the real world is currently in the process to become part of the Web by the movement of the Internet of Things. Interesting is to see the upcoming research of Personalization together with the Internet of "all" Things, real or virtual, alive and inanimate.

ACKNOWLEDGMENTS

This work was partially supported by the European 7th Framework Program project GRAPPLE ("Generic Responsive Adaptive Personalized Learning Environment"): http://www.grapple-project.org.

REFERENCES

[1] Kröner, A.; Schneider, M.; Mori. J. 2009. A Framework for Ubiquitous Content Sharing. IEEE Pervasive Computing, Vol. 8, No. 4, pp. 58-65. IEEE Press.

[2] Wahlster, W., Kröner A., Schneider, M., and Baus, J. 2008. Sharing Memories of Smart Products and Their Consumers in Instrumented Environments. In it – Information Technology, Vol. 50(1), Special Issue on Ambient Intelligence, pp. 45-50, Oldenburg

[3] Heckmann,D. , Schwartz,T., B. Brandherm, M. Schmitz, and M. von Wilamowitz- Moellendorff. GUMO - The General User Model Ontology. In Proc. of Int. Conf. on User Modeling, Edinburgh, UK, 428–432, 2005.

[4] Kröner, A, Heckmann, D, & Wahlster, W., 2006. SPECTER: Building, Exploiting, and Sharing Augmented Memories. In K. Kogure (Ed.), *Workshop on Knowledge Sharing for Everyday Life 2006 (KSEL06)* (pp. 9-16). Kyoto, Japan.

[5] Heckmann, D. (2003b). Introducing situational statements as an integrating data structure for user modeling, context-awareness and resource- adaptive computing. In *ABIS2003*, pages 283-286, Karlsruhe, Germany.

[6] Heckmann, D. (2003a). Integrating Privacy Aspects into Ubiquitous Computing: A Basic User Interface for Personalization. In Krüger, A. and Malaka, R., editors, Artificial Intelligence in Mobile Systems (AIMS 2003), pages 106-110, Seattle, USA. in conjunction with the Fifth International Conference on Ubiquitous Computing.

[7] Schneider, M., Kröner, A. "The Smart Pizza Packing: An Application of Object Memories," *Proc. 4th Int'l Conf. Intelligent Environments* (IE 08), Inst. Eng. and Technology, 2008.

[8] Heckmann, D. Matthias Loskyll, Rafael Math, Pascal Recktenwald, Christoph Stahl. UbisWorld 3.0: a Semantic Tool Set for Ubiquitous User Modeling, Demonstration description in online proceedings of First International Conference on User Modeling, Adaptation, and Personalization (UMAP 2009)

[9] Plate C. et al., "Recomindation: New Functions for Augmented Memories," *Proc. Adaptive Hypermedia and Adaptive Web-Based Systems* (AH 06), Springer, 2006, pp. 141–150.

How Predictable Are You?
A Comparison of Prediction Algorithms for Web Page Revisitation

Ricardo Kawase, George Papadakis, Eelco Herder
L3S Research Center
Hannover, Germany
{kawase, papadakis, herder}@L3S.de

Abstract

Users return to Web pages for various reasons. Apart from pages visited due to backtracking, users typically monitor a number of favorite pages, while dealing with tasks that reoccur on an infrequent basis. In this paper, we introduce a novel method for predicting the next revisited page in a certain user context that, unlike existing methods, doesn't rely on machine learning algorithms. We evaluate it over a large data set comprising the navigational activity of 25 users over a period of 6 months. The outcomes suggest a significant improvement over methods typically used in this context, thus paving the way for exploring new means of improving user's navigational support.

1 Introduction

Nowadays, millions of people browse the Web every second, navigating from site to site and producing massive amounts of navigational log data. These data have the intrinsic potential to provide a solid basis for understanding individual user's behavior. Modeling users is the first step towards this direction, serving as a foundation for developing recommendation and prediction techniques.

Many applications can benefit from effective methods of user modeling, like Web search and personalization/recommendation systems, to name but a few. For example, predictive models have improved the ranking of web search engine results, by computing the distribution of visits over all WWW pages and using it for re-weighting and re-ranking relevant web pages. Navigational information are actually considered more important than text keywords. Hence, the more accurate the predictive models, the better the search results [Brin and Page, 1998].

Several researchers have undertaken the task of understanding user's surfing behavior, by exploiting user data [Adar et al., 2008; Obendorf et al., 2007]. Some go further, using log data to improve algorithms that predict future requests [Awad et al., 2008 ; Gery and Haddad, 2003], while others apply these algorithms to provide users with improved tools for recommendations, bookmarks and history [LeeTiernan, 2003; Pedersen et al., 2010].

As a common practice of studying the past in order to define the future, in this paper we analyze the browser's log data of 25 users along 6 months with a total of 137,737 page requests. Our detailed, statistical analysis of the navigational data gives insights for our research as well as for future work in the area. In addition, we demonstrate a novel user modeling method for predicting the next page that will be revisited. We tested our model on the data set at hand, with the experimental results demonstrating a significant improvement in the support of Web page revisitation over existing methods, commonly used in this area.

2 Related Work

Several past works have explored surfing behaviors with respect to revisitation activity. Although they vary in their estimations, they all recognize that revisitation constitutes a large part of the Web activity. [Herder, 2005], for instance, quantifies it to 50% of the overall Web traffic, while [Cockburn and McKenzie, 2001] approximates it to 80%. They also noted that bookmarks, the most popular revisitation supporting tool, invariably involve managing and organizational problems due to the constantly increasing size of their collections.

Analysis of revisitation. [Tauscher and Greenberg, 1997] describes two important characteristics of revisitation: first, most page revisits pertain to pages accessed very recently; the probability for a page to be revisited decreases steeply with the number of page visits since the last visit. Second, there is a small number of highly popular pages that are visited very frequently; the probability for a page to be revisited decreases steeply with its popularity ranking.

Revisitation behavior has been distinguished by [Obendorf *et al*., 2007] into three different sets: *short-term* (i.e., backtrack or undo within the same session), *medium-term* (i.e., re-utilize or observe a resource in a period of time up to few weeks after the first encounter), *and long-term revisits* (i.e., rediscover a resource several months after the first encounter). The authors further argue that the back button is the most commonly used tool for short-term revisit. For medium-term revisits, the page address is directly typed into the address bar, making use of the automatic URL completion function. However, revisits to a broad range of pages that are accessed on a less frequent basis (i.e., long-term revisits) are poorly supported; users often do not remember the exact address, and ironically browsers do not 'remember' the address either.

[Adar *et al*., 2008] demonstrates that short-term revisits involve hub-and-spoke navigation, visiting shopping or reference sites or pages on which information was monitored. Medium-term revisits pertain to popular home pages, Web mail, forums, educational pages and the browser homepages. As for long term revisits, they involve the use

of search engines, as well as weekend activities (e.g., going to the cinema).

Revisitation Prediction. In [Gery and Haddad, 2003], the authors exploit three methods of Web usage mining: association rules, frequent sequences, and frequent generalized sequence. *Association Rules (AR)* are well documented in the literature as a method that effectively identifies pages that are typically visited together in a same session, but not necessarily in the same order. [Agrawal *et al*., 1993; Agrawal and Srikant, 1995]. *Frequent Sequence Mining* can be considered as equivalent to association rule mining over temporal data sets, while *Frequent Generalized Sequence* introduces sequences that allow wildcards, constituting a more flexible means of modeling users's navigational activity [Gaul and Schmidt-Thieme, 2000]. Their evaluation shows that plain Frequent Sequence Mining performs better in revisitation prediction. However, their dataset consists of server side logs of 3 different websites, thus covering a limited number of possible revisited pages. Contrariwise, in our work we employ browsers's log data to analyze and predict users's, with the set of revisited web pages potentially involving the whole Web.

In [Awad *et al*., 2008], the authors apply two well-established classification techniques in the context of Web surfing prediction: Markov model and Support Vector Machines (SVM). They also combine them in a hybrid method under Dempster's rule and the outcomes of their evaluation suggest that it outperforms the individual methods, especially when domain knowledge is incorporated into it.

3 Data Set and Revisitation Statistics

In this section we briefly introduce the data set that we used for our experiments. We also illuminate the most important aspects of users's revisit behavior – general characteristics as well as individual differences – which are used as a basis for the predictive methods that are evaluated in this paper.

3.1 Data set

The participant pool of our data set consists of 25 participants, 19 male and 6 female. Their average age is 30.5, ranging from 24 to 52 years. The participants were logged for some period between August, 2004 and March, 2005. The average time span of the actual logging periods was 104 days, with a minimum of 51 days and a maximum of 195 days. Participants were logged in their usual contexts - 17 at their workplace, 4 both at home and at work, and 4 just at home.

During the logging period, 152,737 page requests were recorded. 10.1% of them were removed, as they were artifacts (advertisements, reloads, redirects, frame sets). Hence, in total we have 137,737 page requests available for analysis.

3.2 Revisitation Statistics

We recorded an average revisitation rate of 45.6%. Note that this number is lower than in earlier studies, due to the fact that we took into account both GET and POST parameters. The wide range of individual revisitation range (between 17.4% and 61.4%) suggests that revisitation behavior is heavily influenced by personal habits, private interests and the sites visited (for more details, [Obendorf,

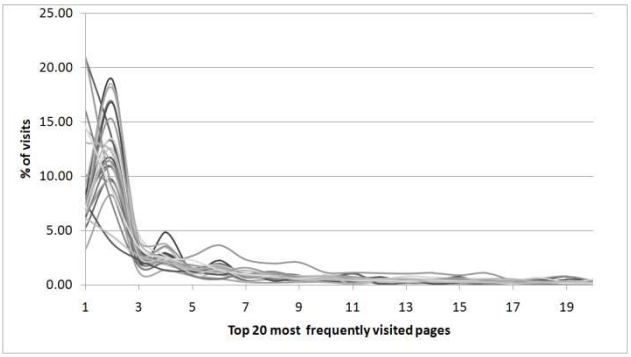


Figure 1: Distribution of most frequently visited pages for each user.

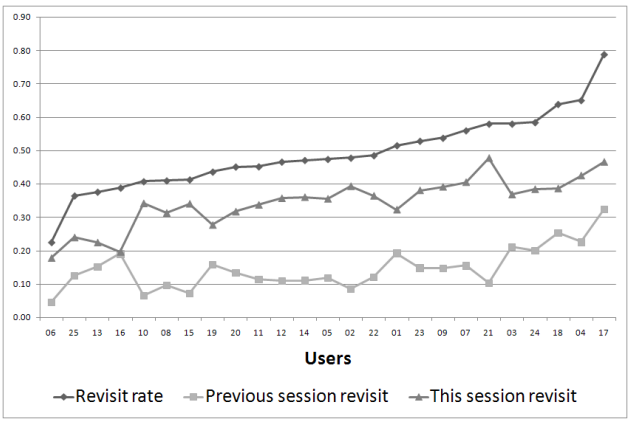


Figure 2: Backtracking and routine behavior plotted against the revisit rate (order by revisit rate).

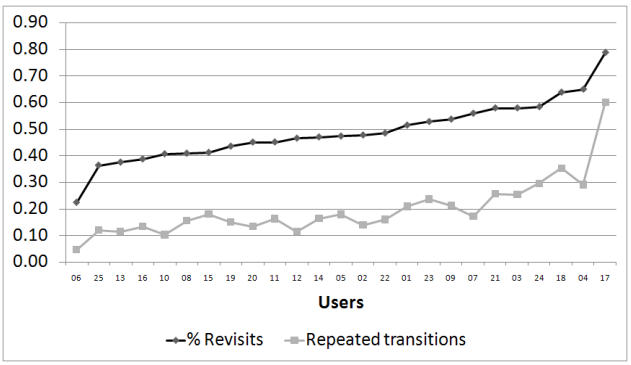


Figure 3: Repetitive behavior (% repeated actions) plotted against the revisit rate.

2008]). In this section we concentrate on individual ferences between users in their *revisitation profile*.

As discussed in Section 2, several studies have identified regularities in revisitation behavior. Users typically have a small set of frequently visited pages, including for example the browser's home page, search engines, favorite news sites, and social networking sites. As can be observed in **Figure 1**, the distribution of *most frequently used* pages clearly follows a power law for most of the users, but not for all – some have a large number of pages in their browsing routine.

The distribution of revisits to pages based on the number of pages between the last visit and the current visit does follow a power law distribution for all users. Consequently, the *backtracking activities* (revisits to pages in the current session) and *routine behavior* (revisits to pages in previous sessions) grow roughly linear with the revisit rate. This illustrated in **Figure 2** - note that despite the

correlation there are still users that can be identified as predominantly backtrackers or predominantly routine revisitors. The average percentage of backtracking actions among revisits is 75%, with a minimum of 51% and a maximum of 84%.

Predictive models of Web navigation, such as Markov models, typically assume that users exhibit a rather large percentage of *repetitive behavior*, including sequences of pages that are regularly visited in the same order. In **Figure 3** we plot the users' repetitive behavior (based on the ratio between the number of unique pairs of pages that a user visited consecutively and the total number of transitions). The average percentage of repetitive transitions is 20%, with a minimum 5% and a maximum of 60%.

3.3 Discussion

Based on the statistics in the previous section, it becomes clear that Web page revisitation behavior follows sufficient regularities to be exploited for enhanced revisitation support – in a similar manner as is already common in recommender systems based on Web usage mining and collaborative filtering. Earlier work on revisitation (see Section 2) confirms this observation, but only to a limited extent.

In this following, we investigate, compare and combine the performance of several predictive methods for page revisitation. Our analysis attempts a general comparison of several prediction mechanisms, with the aim of identifying the best performing one, knowing though that their performance depends heavily on the regularities in the individual user's revisitation activities.

4 Prediction of Next Page Visits

The problem we are tackling in this paper can be formally defined as follows:

Given a collection of Web Pages, $\boldsymbol{P = \{p_1, p_2, ...\}}$*, that have been visited by a user,* $\boldsymbol{u}$*, during his past* $\boldsymbol{n}$ *transactions,* $\boldsymbol{T_u=\{t_1, t_2,\ldots, t_N\}}$*, rank them so that the ranking position of the page re-visited in the next,* $\boldsymbol{n+1}$*, transaction is the highest possible.*

The methods copying with this problem should exclusively try to facilitate the revisitation of already accessed pages , rather than trying to suggest to a user new pages that seem relevant to his surfing activity. The ranking of all web pages is updated after every transaction, and the higher the ranking position of the subsequently accessed page, the better. In fact, the lowest possible Average Ranking Position (**ARP**) of revisited pages, the higher the performance of the algorithm. This is in line with the intuition behind ranking search engine's query results: the higher the ranking of the desired resource, the better the performance of the search engine [Brin and Page, 1998].

To solve the aforementioned problem, we employ a framework combining two categories of methods. The first one involves *ranking methods*: they estimate for each web page the likelihood that it will be accessed in the next transaction based on some evidence, such as the recency or the frequency of earlier visits to this page. The second category covers *propagation methods*; these are techniques that capture repetitiveness in the surfing behavior of a user and identify groups of pages that are typically visited together, in the same session but not necessarily in a specific order.

In the following, we provide a brief outline of our framework that conglomerates these two categories of prediction methods. The implementation of the methods presented here is publicly available through the SUPRA project of SourceForge.net[1].

4.1 Ranking Methods

The aim of ranking methods is to provide for each web page a numerical estimate of the likelihood that it will be accessed in the next transaction. In this work, we consider the following ranking methods:

1. Least Recently Used (**LRU**)
2. Most Frequently Used (**MFU**)
3. Polynomial Decay (**PD**)

The first two methods, namely LRU and MFU, constitute well-established caching algorithms that are typically employed in prediction tasks. LRU is based on the idea that the more recently a web page was visited, the most likely that it will be re-visited in the immediate future. Hence, it assigns the highest ranking position to the latest accessed page. MFU, on the other hand, relies on the idea that the more often a web page is visited, the most likely it is to be revisited in the next transaction.

[Papadakis et al., 2010] demonstrated, though, that these methods are not adequate for effectively predicting future revisitations on server-side logs of closed corpus websites. Due to their unidimensionality, LRU produces a plainly chronological arrangement of web pages based on their recency, while MFU takes into account merely their *degree of usage*. More accurate predictions can be achieved when incorporating both evidences into a single, comprehensive method.

To this end, [Papadakis et al., 2010] introduced the decay ranking model for predicting the next revisited page. According to this model, the value $\boldsymbol{v_{in}}$ of a web page $\boldsymbol{w_i}$ after $\boldsymbol{n}$ transactions $\boldsymbol{T_u}$ of user $\boldsymbol{u}$ is derived from the following formula:

$$v_i = \sum_{k=0}^{n} d(t_k, w_i, n), \text{ where}$$

$\boldsymbol{d(t_k, w_i, n)}$ is a ***decay function*** that takes as an input the *k-th* transaction, t_k, of user, *u*, together with the index of the current transaction, *n*, and gives as output the value of this transactions for web page w_i . Every valid decay function should satisfy the following *properties* ([Cormode et al., 2009]):

1. $d(t_k, w_i, n) = 1$ when k=n
2. $d(t_k, w_i, n) = 0$ if t_k doesn't pertain to web page w_i
3. $0 \leq d(t_k, w_i, n) \leq 1 \; \forall \; 0 \leq k \leq n$
4. d is monotone non-increasing as n increases ($0 \leq k \leq n$):

$$n' \geq n \rightarrow d(t_k, w_i, n') \leq d(t_k, w_i, n)$$

Among the various decay function families that satisfy these properties, the *polynomial decay functions* were found to outperform both the *exponential* and the *logarithmic* ones. The reason is that their smooth decay balances harmonically the recency and the degree of usage of web pages; in contrast to this, exponential functions convey a steep decay that puts more emphasis on the recency of usage, whereas the logarithmic functions promote excessively the degree of usage, due to their excessively slow decay.

The form of a ***polynomial decay function with exponent*** $\boldsymbol{\alpha}$ is the following:

[1] http://sourceforge.net/projects/supraproject/

$$d(t_k, w_i, n) = \frac{b}{1+(n-k)^{\alpha}}, \text{ where}$$

$\boldsymbol{b}$ is equal to 1 if t_k pertains to w_i, and 0 otherwise.

4.2 Propagation Methods

Unlike ranking methods that produce an ordering of web pages, propagation methods aim at detecting and capturing patterns in the surfing activity of users. They identify those pages that are commonly visited within the same session and associate them with each other. The "links" created by these methods can be combined with a ranking method, so that the value of a web page is propagated to its relevant ones. In this way, the higher the value of a web page, the more the pages associated with it are boosted and the higher their ranking position.

In this work, we distinguish between two families of propagation methods: *those that take into account the order of the transactions within a session, and those who disregard this order*. For the former case, we consider transition matrices, whereas for the latter we examine association matrices.

4.2.1 Transition Matrix

Similar to a first-order Markov model, a transition matrix (**TM**) is a two dimensional structure with its row and columns representing the enumeration of web pages; each cell *TM(x,y)* expresses the number of times that a user visited page *y* after *x*. Given that a transition matrix respects the order of accesses within a session, it is not a symmetrical one: the value of *TM(x,y)* is not necessarily equal to that of *TM(y,x)*. Moreover, its diagonal cells are all equal to 0: $\mathrm{TM}(x,x) = 0\ \forall x$.

In the following, we introduce 4 different approaches to correlating web pages according to the past navigational activity in order to build the transition matrix. They can be intuitively illustrated through a simple walkthrough example. Given a set of 4 web pages – *A*, *B*, *C*, *D* - and the following set of transactions during a user session

A ⇨ B ⇨ C ⇨ D ⇨ A

we can associate these pages in four ways (taking into account the order of the accesses):

1. **Simple connectivity** – For each transition *x*→*y* in the given session, only the value of the cell *TM(x,y)* is incremented by one. **Figure 4a)** depicts the values of the transition matrix according to the simple connectivity rule after the last transition of the given session *D*→*A*.
2. **Continuous connectivity** – Each web page visited within the current session is associated with all the subsequently accessed pages. In our example, after transition *D*→*A*, *A* is associated with all other web pages (*B,C,D*) incrementing the corresponding cells by one, as shown in **Figure 4b)**.
3. **Decreasing continuous connectivity** – This strategy operates in a similar way as the previous one (i.e., connecting all the pages within a session) with the difference that it adds a decay parameter representing the distance (i.e., number of transitions) that intervene between two web pages. In our example, cell *(C,A)* is incremented by ½ after *D*→*A*, since page *C* is two steps away from the page A. **Figure 4c)** depicts the values of the transition matrix according to the decaying continuous connectivity rule after the transition *D*→*A*.
4. **Increasing continuous connectivity** – Is the inverted version of the previous strategy. Instead of decreasing the additional value of cell *TM(x,y)* according to the distance of pages *x* and *y*, it increases it proportionally. The outcomes of this rule after transition *D*→*A* are presented in **Figure 4d)**.

It is worth noting that the simple connectivity transition matrix was also used in [Awad *et al*., 2008], but its frequencies were used as features of a classification algorithm instead.

(a)

	A	*B*	*C*	*D*
A	0	1	0	0
B	0	0	1	0
C	0	0	0	1
D	1	0	0	0

(b)

	A	*B*	*C*	*D*
A	0	1	1	1
B	1	0	1	1
C	1	0	0	1
D	1	0	0	0

(c)

	A	*B*	*C*	*D*
A	0	1	½	¼
B	¼	0	1	½
C	½	0	0	1
D	1	0	0	0

(d)

	A	*B*	*C*	*D*
A	0	1	2	4
B	4	0	1	2
C	2	0	0	1
D	1	0	0	0

Figure 4: Transition matrix example.

4.2.2 Association Matrix

In contrast with transition matrices, association matrices (**AM**) are based on the idea that the temporal order of transactions within a session is not important; pages that are visited in the course of the same session should be equally connected with each other, regardless of their order and the number of transitions that intervene between them. The rationale behind this idea is that users may visit a group of pages XYZ on a regular basis, but not necessarily in that order.

In this context, an association matrix is built simply by associating all the pages that are visited in a single session. Given the session presented above, the resulting AM has all non-diagonal cells equal to one, as all resources were accessed during this session (**Figure 5**).

A variation of the association matrix can be derived by normalizing its values with the help of the mutual information. More specifically, this involves the multiplication of each cell *TM(x,y)* of AM with the following mutual information factor (*mif*):

$$mif(x,y) = AM(x,y) \cdot \log \frac{p(x,y)}{p(x) \cdot p(y)}, \text{ where}$$

- *AM(x,y)* is the number of sessions containing both page x and y (i.e., the value of the cell *AM(x,y)*),
- *p(x,y)* is the probability of a session to contain pages *x* and *y* (i.e., the value of *AM(x,y)* divided by the number of sessions)

- *p(x)* (*p(y)*) is the probability that a session contains page *x* (*y*) (i.e., the number of sessions with x or y divided by the total number of sessions).

Without this smoothing factor, the values of AM are biased towards pairs of pages that have a high frequency of co-occurances, although they are not highly correlated.

$$\begin{array}{c} \\ A \\ B \\ C \\ D \end{array} \begin{array}{c} \begin{array}{cccc} A & B & C & D \end{array} \\ \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

Figure 5: Association matrix example.

4.3 Combining Ranking with Propagation methods

To combine the available ranking methods with the variations of the propagation techniques, we employ a simple, linear scheme: following a transaction, the value of each web page is first (re)computed, according to the selected ranking method. Then, for each non-zero cell of the transition matrix at hand, *TM(x,y)*, we increase the value of page y, v_y, as follows:

$$v_y += p(x \to y) \cdot v_x, \text{ where}$$

- *p(x→y)* is the transition probability from page x to page y, estimated by $p(x \to y) = \frac{TM(x,y)}{\sum_i^N TM(x,i)}$, and
- v_x is the value of x estimated the ranking method.

In case an association matrix is used as a propagation method, the v_y is increased as follows:

1. $v_y += AM(x,y) \cdot v_x$ for the plain association matrix, or
2. $v_y += mif(x,y) \cdot v_x$ for the mutual-information-normalized association matrix.

All in all, considering the 3 ranking methods alone and in combination with the 4 variations of TM and the 2 variations of AM, we have 21 distinct ranking methods. Due to space limitation and for the sake of readability, the following, evaluation section focuses merely on the best performing ones.

5 Evaluation Setup and Discussion of Results

5.1 Setup

To evaluate experimentally our framework of methods, we employed the data set described in section 3, comprising 25 distinct users and 137,737 page requests in total (not evenly distributed among the users). In more detail, we simulated the navigational activity of each user, independently of the others. After each transaction, the ranking of all visited pages was updated, and, in case the next access was a revisitation, the position of the corresponding web resource was recorded. Having all these ranking places for each recommendation method, we derived the following metrics for evaluating its performance:

- *Precision at 10* (***P@10***): it expresses the percentage of revisitations that involved a web page ranked in some of the top 10 positions. The higher this percentage, the better the performance of the recommendation method. This metric provides evidence for the usability of the prediction method, as users typically have a look only at the first 10 pages presented to them (just like they do with web search engine results).
- *Average Ranking Position Reduction Ratio (**RR**)*: it denotes the degree of improvement conveyed by the prediction method in comparison with the actual revisititation behavior of the user. More specifically, it is computed from the following formula:

$$RR = \frac{AcARP - PrARP}{AcARP} \cdot 100\%, \text{ where}$$

  - *AcARP* is the *Ac*tual *A*verage *R*anking *P*osition of the user, representing the average distance in terms of the number of page requests that intervene between the revisited web pages, and
  - *PrARP* is the *Pr*ediction *A*verage *R*anking *P*osition, expressing the place a revisited page is found on average in the ranking list that the prediction method produces.

The higher the value of RR, the better the performance of the recommendation algorithm, with negative values denoting that PrARP is lower than AcARP (i.e., no improvement with respect to the actual revisitation behavior of the user). RR provides, thus, an estimation of the overall performance of a prediction method, since it considers the performance over all the revisitations in the navigational history of a user, and not only the highest ranked ones.

On the whole, the combination of these two metrics provides a comprehensive estimation of the effectiveness of a recommendation algorithm in predicting the next revisited page; they cover both the recommendations that are indeed useful for users as well as their performance in all the cases.

5.2 Results analysis

Regarding the performance of the ranking algorithms we are considering, it is summarized in **Figure 6**, with the RR depicted in **Figure 6a)** and the P@10 in **Figure 6c)**. It is evident that the baseline MFU performs much worse than the other methods. This is explained by the fact that backtracking (LRU) is more common than revisiting popular sites, thus ensuring much higher performance for LRU. Our proposed method, the Polynomial Decay, which is a combination of MFU and LRU, exhibits the best performance for all users, improving in each case that of LRU to a varying but considerable extent.

The performance of PD is significantly enhanced when combined with AM and TM, with TM accounting for a higher improvement. This is the case with respect to both metrics, as is clearly depicted in **Figure 6b)** for RR and **Figure 6d)** for P@10. Conversely, the combination of LRU and MFU with AM and TM results in a lower performance for both metrics (that's why their performance is not included in the figures). This suggests that users do not have many regular patterns in their page visit behavior (i.e. after having visited page X they do not always visit page Y). It is interesting to note, though, that PD achieves by far the best results in combination with the Simple TM, while LRU and MFU are better combined with the Increasing and the Decreasing TM, respectively.

Another observation is that, despite the different assumptions that lie behind the algorithms, there is a corre-

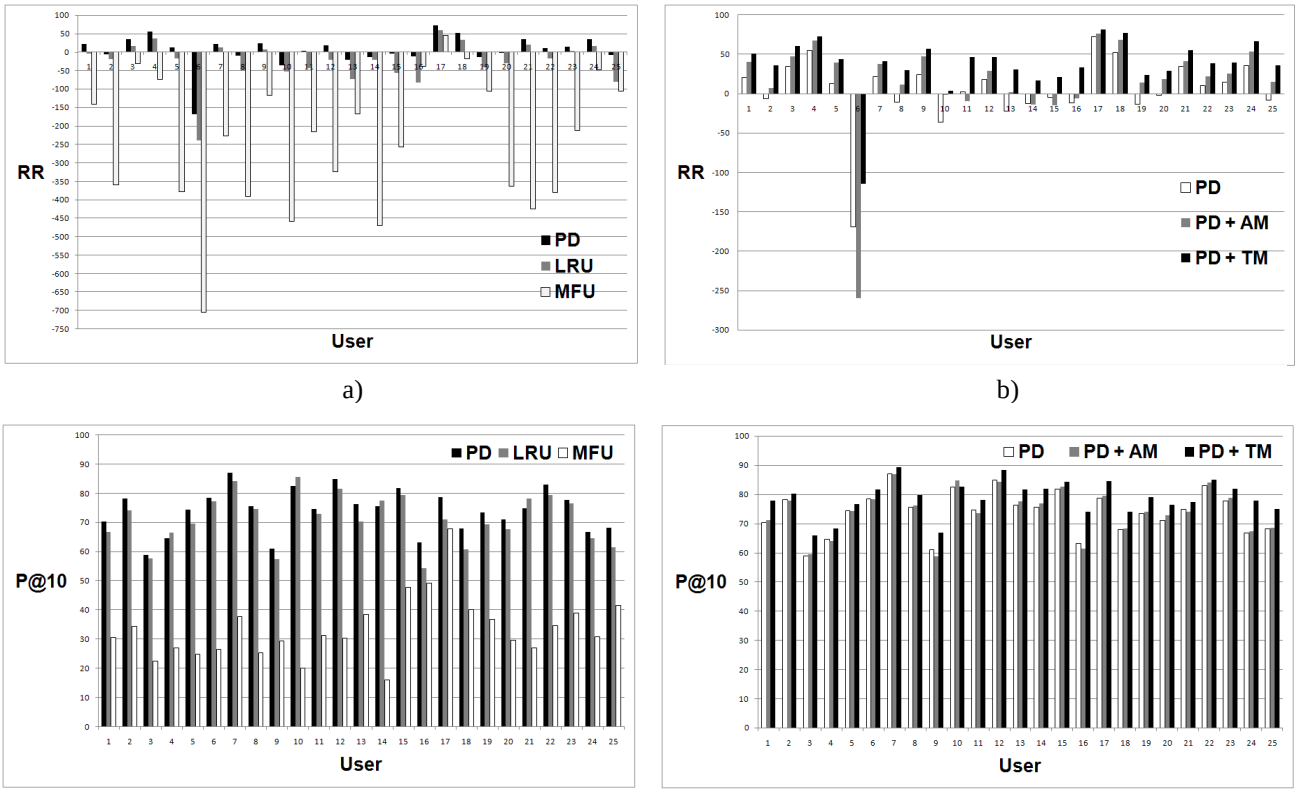


Figure 6: a) Reduction ratios of the average ranking position for LRU, MFU and PD. b) PD with AM and TM(simple. c) Precision at 10 for LRU, MFU and PD. d) PD with AM and TM(simple) on the bottom right.

lation between the performances of the algorithms per user. This can be observed in **Figure 7**, where the better the performance of the best-performing algorithm, PD+TM, the better the performances of PD, PD+AM and LRU. From the same figure it also becomes clear that there is no correlation at all with the revisit rate: one would expect that users who revisit pages more often – who are shown to have more frequent transitions – are more predictable in their behavior; this turns out not to be the case. Note also that the – poor – performance of MFU does not follow the pattern of the other algorithms and is not correlated with the revisit rate either.

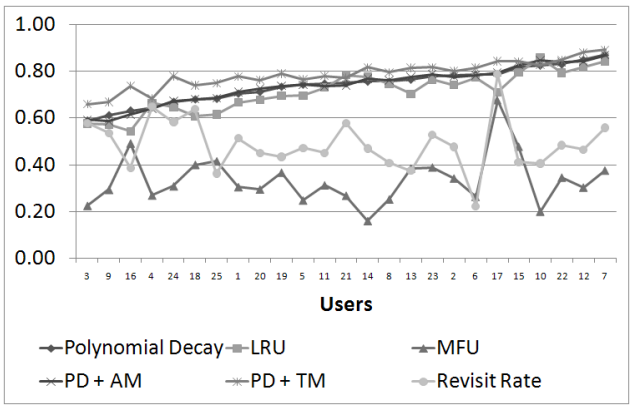


Figure 7: Performance (P@10) of the different algorithms per user. Users are ordered by the best-performing algorithm.

5.3 Discussion

In our analysis we compared various algorithms and combinations of algorithms for predicting which pages users will revisit in a session. These algorithms exploit the following characteristics of revisits:

- Revisits are typically focused on pages visited very recently.
- The more revisits, the more repetitive behavior in terms of transitions between pages.
- There is a small group of pages that is visited very frequently.

It turns out that, even though a small set of frequently visited pages covers the majority of revisits, the recency effect, as well as frequent transitions, plays a larger role in the prediction algorithms.

The evaluated algorithms, in particular Polynomial Decay in combination with the transition matrix, significantly improve upon the list of most recently used pages, in particular for users whose the list of LRU pages performs relatively bad. The differences become smaller together with the increase of the recency effect.

The counter-intuitive effect that a higher recency rate does *not* lead to better predictions can be explained by the many differences in individual behavior between users (such as the number of news sites or bulletin boards that a user actively follows, strategies for search and backtracking, the number of reoccurring activities).

From this we can conclude that revisiting behavior is mainly influenced by the recency effect, but it definitely makes sense to take the popularity of pages and the currently/last visited page (the user's current context) into account as well.

6 Conclusions

In this paper we studied the browsing behavior of 25 users during a period of approximately 6 months. We analyzed the data to build a comprehensive stereotype of users' behavior, focusing on their revisitation patterns. We also ran experiments applying a variety of methods to predict users's revisitation.

Our proposed Polynomial Decay algorithm in combination with users's navigational patterns as they are encapsulated by the Transition Matrix outperforms substantially existing methods commonly used for revisitation prediction.

In our previous work [Papadakis et al., 2010] we demonstrated the better performance of our methods on a server-side dataset. Combining with the results of the work presented here (on a client-side dataset) we can firmly claim that our proposed method is more effective than the baselines LRU and MFU for both cases.

Though the experiments presented here are on the field of Web Usage Mining, our real goal is to improve the support of revisitation by the means of intelligent user interfaces. Hence, this was the first step towards a more effective user modeling method. Our plan, as future work, is to implement a browser interface that allows users to interact with the output of our methods: a collection of related URLs that does not contain only the obvious selections, but also related websites that are usually overlooked between the head and the long tail.

References

[Adar *et al*., 2008] Adar, E., Teevan, J., and Dumais. S. T: Large scale analysis of Web revisitation patterns. In CHI, pages 1197-1206, 2008.

[Agrawal *et al*., 1993] Agrawal, R., Imielinski, T., and Swami, A. N:. Mining association rules between sets of items in large databases. In SIGMOD, pages 207–216, 1993.

[Agrawal and Srikant, 1995] Agrawal, R., and Srikant, R.: Mining sequential patterns. In ICDE, pages 3–14, 1995.

[Awad *et al*., 2008] Awad, M., Khan, L., and Thuraisingham, B.:Predicting WWW surfing using multiple evidence combination. In The VLDB Journal 17, 3, pages 401-417, 2008.

[Brin and Page, 1998] Brin, S., and Page, L.: The anatomy of a large-scale hypertextual web search engine. In Computer Networks 30(1-7), pages 107-117, 1998.

[Cockburn and McKenzie, 2001] Cockburn, A., and McKenzie, B.: What do Web users do? An empirical analysis of Web use. In Int. J. of Human-Computer Studies, 54(6), pages 903-922, 2001.

[Cormode *et al.*, 2009] Cormode, G., Shkapenyuk, V., Srivastava, D., and Xu, B.: Forward Decay: A Practical Time Decay Model for Streaming Systems, In ICDE, pages 138-149, 2009.

[Gery and Haddad, 2003] Gery, M., and Haddad, H.: Evaluation of web usage mining approaches for user's next request prediction. In WIDM, pages 74–81, 2003.

[Gaul and Schmidt-Thieme, 2000] Gaul, W., and Schmidt-Thieme, L.: Mining web navigation path fragments. In WEBKDD, 2000.

[Herder, 2005] Herder, E.: Characterizations of user Web revisit behavior. In Proceedings of Workshop on Adaptivity and User Modeling in Interactive Systems, 2005.

[LeeTiernan, 2003] LeeTiernan, S., Farnham, S., and Cheng, L.: Two methods for auto-organizing personal web history. In CHI Extended Abstracts on Human Factors in Computing Systems, pages 814-815, 2003.

[Obendorf *et al*., 2007] Obendorf, H., Weinreich, H., Herder, E., and Mayer, M.: Web page revisitation revisited: Implications of a long-term click-stream study of browser usage. In CHI, pages 597-606, 2007.

[Papadakis et al., 2010] Papadakis, G., Niederee, C., Nejdl, W.: Decay-based Ranking for social application content. In WEBIST, 2010.

[Pedersen et al., 2010] Pedersen, E. R., Gyllstrom, K., Gu, S., and Hong, P. J.:Automatic generation of research trails in web history. In IUI, pages 369-372, 2010.

[Tauscher and Greenberg, 1997] Tauscher, L., and Greenberg, S.: How people revisit Web pages: Empirical findings and implications for the design of history systems. InInternational Journal of Human-Computer Studies, v.47, n.1, pages 97-137, 1997.

User and Document Group Approach of Clustering in Tagging Systems

Rong Pan, Guandong Xu and Peter Dolog
IWIS — Intelligent Web and Information Systems
Department of Computer Science
Aalborg University
{rpan, xu, dolog}@cs.aau.dk

Abstract

In this paper, we propose a spectral clustering approach for users and documents group modeling in order to capture the common preference and relatedness of users and documents, and to reduce the time complexity of similarity calculations. In experiments, we investigate the selection of the optimal amount of clusters. We also show a reduction of the time consuming in calculating the similarity for the recommender systems by selecting a centroid first, and then compare the inside item on behalf of each group.

keywords: User Profile, Document Profile, Spectral Clustering, Group Profile, Modularity Metric

1 Introduction

The success of social tagging resulted in the proliferation of sites like Delicious, CiteUlike, Digg, or Flickr. Such sites contain large amount of user tagged data for information retrieval in social-tagging systems [6; 7; 12; 16; 17], or for the establishment of user profiles and the discovery of topics, among other applications.

[5] uses the tags associated with specified objects to build a single user profile. However, here comes a problem: it is hard to express the entire user profile or the document profile. The traditional user profile expresses the users' preferences depending on collecting users' behaviors information, such as provides many tedious options in their registrations. The disadvantage with such an approach is too much reliance on users who is not able very often to express his entire user profile and interests. The document profile shows the background, categories, and keywords, it also depends on the description when it is added into the system. However, with the increase of the number and types of users, it's hard to express the different emphases for various users with the same document.

In social collaborative (tagging) systems, the common perception or judgment on documents are determined by a group of users rather than a single user. In a similar way, by using a group of documents, rather than one document, it might represent much more specific information during the information search.

Therefore, our assumption is that by utilizing the community views of users and documents, we are able to facilitate the organization of information resources in search and navigation.

In social tagging systems, users express their judgment by annotations or tagging. The tag can endorse their opinions on various web items, which is one of the defining characteristics of Web 2.0 services, allowing them to collectively classify and index information for later search and sharing. With social tagging, a user can express his own perspective on web items, e.g. resources like images, videos, scientific papers, thus allowing other like-minded users to find and use the similar information.

The tagging has been already utilized for organizing the resources. [3] develops a page rank algorithm of resources based on preference tag vectors. [6; 8; 7] investigate social and behavioral aspects of a tag-based recommender system which suggests similar web pages based on the similarity of users' tags. However, there is another problem emerging: not all of the social tagging systems proposed so far maintain high quality and quantity of tag data. It is particularly prominent when a new user enters the system or a new document is added into the system.

If the individual user profile or document profile can be collected and grouped into several groups characterized by the significant tags, it is believed that common tags annotated by the most objects inside the group can reflect the characteristics of user preference or document functionality. Moreover, it will be of benefit for solving the problem of low tag quality of individual user or document. Even when a new user or a new document is added into the system, the tags can be extended to the user by referring to the majority tagging behavior of users on documents.

Regarding to the previous problems, even if the tag is rich enough for the users and documents, the time consuming is still very high when a user wants to get the most appropriate document from a large document database, since the system has to calculate the similarity between users or documents one by one.

We propose the method that calculates the similarity between the target tag vector and the centroids of all clusters to determine the cluster with highest similarity, then calculate the similarity of the target tag vector with the document profiles inside the cluster to rank the whole documents. In such way the time consuming can be reduced. Since we have got the groups of user profile or document profile, how to choose the number of clusters is another problem. The traditional way is to assign the initial clustering number manually. In this paper, we use the modularity metric [13] to evaluate the optimal number for the clusters.

Based on the problems mentioned above, this paper

proposes an approach for group modeling by utilizing a clustering algorithm. The group modeling aims at assigning the individual users or document profiles into different groups, which correspond to various user preferences or content relatedness from the large amount of data for tagging.

User Group Profile and *Document Group Profile* can be generated from individual user profiles and document profiles; both of them are expressed by the tags. Group profiling is not constructed based on stereotypes but based on the results of clustering algorithms from transactional data. It can identify the objects inside the community with similar tags, and collect the data for the similar objects. It can expand the tag set for the individual object inside the community which is helpful for the poor tag quality and quantity. Furthermore, for making tag-based recommendation, it will significantly reduce the time consuming in calculating the similarity between the user and document groups.

The main contributions in this paper are:

1. A group modeling method by utilizing the clustering algorithm.
2. The most appropriate number of clusters to generate the User Group Profile and the Document Profile by using the modularity metric.
3. Reduction in time needed for computation for organizing the documents comparing to the other methods.

The rest of the paper is organized as follows: Section 2 presents the related work in the field of clustering and profiling. In section 3, we describe the preliminaries for the data model. Section 4 discusses the details of user profile and document profile with the introduced mathematical models and how to get the group profile by utilizing the spectral clustering algorithm. The experiment is designed in terms of datasets and evaluation measures in section 5, and experimental results and comparisons are presented in this section as well. We conclude the paper and discuss possible future research directions in section 6.

2 Related Work

The folksonomy in [3] has been defined as a data structure that evolves over time when people annotate resources with freely chosen words. It is user-contributed data aggregated by collaborative tagging systems. In such systems, users are allowed to choose terms freely to describe their favorite web resources. A folksonomy is generally considered to consist of at least three sets of elements, namely users, tags and resources. Although there can be different kinds of resources.

The prerequisite of personalization is to acquire user profile that describes user's interests, preferences and background knowledge about specified domains. Methods are used for modeling user profiles include logic-based representation and inference, Bayesian models, feature-based filtering, Clique-based filtering, and neural networks. However, such user profile is still for individual persons. Our approach is to cluster the similar users in the same communities. [5] proposes to create user profiles from the data available in such folksonomy systems by letting user specify the most relevant objects in the system. Instead of using the objects directly to represent the user profiles, they use the tags associated with the specified objects to build the user profiles.

[20] presents analysis on the personal data in folksonomies, and investigates how accuracy rate user profiles can be generated from this data. They propose an algorithm to generate user profiles which can accurately represent the multiple interests.

F. Durao and P. Dolog in [6; 8; 7] present a tag-based recommender system which suggests similar Web pages based on the similarity of their tags from a Web 2.0 tagging application. They also propose an approach to extend the basic similarity calculus with external factors such as tag popularity, tag representativeness and the affinity between user and tag.

K. R. Bayyapu and P. Dolog in [2] tries to solve the problems of sparse data and low quality of tags from related domains. They suggest using tag neighbors for tag expression expansion. However the tag neighbors are based on the content of documents. We propose another approach to extend the tag set by the group profiling.

[19] uses a framework of User-Profile Modeling based on Transactional data for modeling user group profiles based on the transactional data which can incorporate external information, either by means of an internal knowledge base or on dynamic data supplied by a specific information extraction system. Such user group profiles consist of three types: basic information attributes, synthetic attributes and probability distribution attributes. User profiles are constructed by clustering user transaction data and integrating cluster attributes with domain information extracted from application systems and other external data sources. And Teevan et al. apply group profiles to personalize search by an algorithm to "groupiz" (versus "personalize") in result ranking on group-relevant queries [18] , Abel et al. [1] shows that the quality of search result ranking in folksonomy systems can be significantly improved by introducing and exploiting the grouping of resources and Mei and Church show that group profiles facilitate Web search [11].

Clustering can divide the large amount of data into several groups. Clustering algorithms, specially designed for transactional data, can efficiently partition historic user transactions into clusters [9][15]. Each cluster is a set of transactions representing the interests of a particular user group. It is the assignment of a set of observations into subsets so that observations in the same cluster are similar in some sense. We want to use the clustering for unsupervised learning in the group profiling.

3 Preliminaries for Folksonomy Data Model

The user profile can be used to store the description of the user's characteristics. Such information can be exploited in social tagging systems for taking the persons' characteristics and preferences into account . For example, the social tagging systems usually ask the users to choose their own words as tags to describe the favorite web resource. So the user profiles can justify the benefit and interest for various users.

The document profile is represented by the metadata generated by the community of users tagging the

documents. It is the process that refers to the construction of a profile for a specific via the extraction from a set of tagging data.

When users want to annotate web resource for better organization and use the relevant information to their needs later, they will tag such information with free-text keywords. The tags, which are given by the users, reflect the navigational preference and interest of them. On the other hand, with the increase of documents number that the user visited and annotated, each user has his own tag set which characterizes the interest or preference. Likewise, each tagged document also has its own tag set which expresses the content relatedness and subject of the document. In the context of social tagging systems, the user profiles and document profiles thus are expected to be represented by the representative tags. Therefore the process of user and document modeling is to capture the significant tags from a large volume of tagging data in a social collaborative environment.

There are a number of studies on user and document profiling (see for example [20; 19]). Amongst them, the basic idea of such approaches is originated from the introduction of a specific mathematical modeling of folksonomy. The folksonomy is a three-dimensional data model of social tagging behaviors of users. In social tagging systems, both the user profiles and document profiles are formulated starting from the folksonomy model. In the following section, in order to well reveal the mutual relationships between these three-fold entities, i.e. user, item and tag, we firstly briefly discuss the data model used in the following group profiling processes.

A folksonomy F according to [10] is a tuple $F = (U, T, D, A)$, where U is a set of users, T is a set of tags, D is a set of Web documents, and $A \subseteq U \times T \times D$ is a set of annotations. The relationship is shown as Fig1.

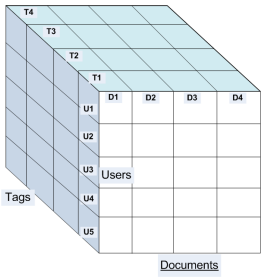


Figure 1: Relationship of users, tags, resources in folksonomy

We can construct the folksonomy data model from the tagging data by such following steps: collecting the data of users, tags and resources from the explicit information and implicit information. And then represent them in the three-dimensional vector space. Based on this we can define the documents' data as the X coordinate, the users' data as the Y coordinate, and tags' data as the Z coordinate. The relationship in folksonomy is $R_{tagging} = U \times T \times D$, $R_{tagging} \in A$, where $U = \{U_1, U_2, ..., U_m\}$ is the set of users and $T = \{T_1, T_2, ..., T_k\}$ is the set of tags,$D = \{D_1, D_2, ..., D_n\}$ is the set of documents. Shown in Fig1, for each point in the three-dimensional vector space, it can be defined as user $u \in U$ has tagged document $d \in D$ with tag $t \in T$.

Upon the folksonomy data model, we can derive the user and document profile by utilizing the relationship among the users, tags and documents in the tagging procedures, which will be discussed in the following section.

4 User Group Profiling and Document Group Profiling by Clustering

As mentioned in the introduction section, the poor quality and quantity of tag data would be a problem. Meanwhile, the time complexity is also a big concern when calculating the recommendation rank for the objects based on the large amounts of data. In the following parts, this paper will focus on solving such problems.

4.1 User Profile and Document Profile

In the social tagging systems, we can get the user profile and document profile by utilizing and analyzing the relationships among the users, tags and documents modeled in folksonomy.

First of all, we discuss the user profiling. For a given user, if we want to study his interests, only the tags, associated with documents, need to be concentrated on. In the folksonomy data model, we can use a user vector assgined with a unique id, for example, the user $U_i \in U, i$= 1,..., M.

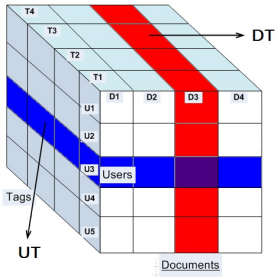


Figure 2: Matrix UT and Matrix DT in folksonomy

As shown in Fig2, a two-dimensional matrix UT_i is extracted from the relationship between the documents and tags for a particular user. In UT_i , each column is corresponding to the documents $D_n \in D, n$= 1,..., N that used by user U_i , and each row is corresponding to the tags $T_k \in T, k = 1, ..., K$.

$$UT_i = \begin{bmatrix} u_{11}, & u_{12}, & \cdots, & u_{1n} \\ u_{21}, & u_{22}, & \cdots, & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ u_{k1}, & u_{k2}, & \cdots, & u_{kn} \end{bmatrix}, u_{kn} \in \{0, 1\}$$

Here u_{kn} means that if there exists an association between tag T_k and document D_n, annotated by user U_i, the u_{kn} sets to 1, otherwise it is 0.

By accumulating the row of matrix UT_i, the frequency of tag is defined as $t_{ik} = \sum_{n=1}^{N} u_{kn}$ which reveals the user's preference and interest. Then we can obtain the full set of the pairs of tags and their frequency weights. So the profile of userU_i in the form of tag set can be defined as UP_i =

$\{(T_1, t_{i1}), (T_2, t_{i2}) \cdots (T_k, t_{ik})\}, k = 1, \cdots, K$, where $t_{ik} = \sum_{n=1}^{N} u_{kn}$, $T_k \in T, k = 1, \cdots, K$.

Similarly, given a document D_n, we can obtain another two-dimensional matrix DT_i, where each column denotes the user U_i and row is the related tags that the userU_i is used to annotate the documentD_n. The size of matrix DT_i is M users by K tags.

$$DT_i = \begin{bmatrix} v_{11}, & v_{12}, & \cdots, & v_{1m} \\ v_{21}, & v_{22}, & \cdots, & v_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ v_{k1}, & v_{k2}, & \cdots, & v_{km} \end{bmatrix}, v_{km} \in \{0, 1\}$$

The elementv_{km} in DT_i is corresponding to the user U_m and tag T_k. The value of v_{km} is defined that, if there exists an annotation between tag T_k and user U_m, that means T_k is associated with the U_m, the v_{km} sets to 1, otherwise it is 0.

By accumulating the row of matrix DT_i, the frequency of tag is defined as$t_{ik} = \sum_{m=1}^{M} u_{km}$ Then we can obtain the full set of the pairs of tags and their frequency weights. So the profile of document D_n in the form of tag set can be defined as $DP_i = \{(T_1, t_{i1}), (T_2, t_{i2}) \cdots (T_m, t_{im})\}, m = 1, \cdots, M$, where $t_{ik} = \sum_{m=1}^{M} u_{km}$, $T_k \in T, k = 1, \cdots, K$.

From the steps mentioned above, the user profiles and document profiles are defined as a single user or document respectively rather than a group of users or documents. However, in social tagging systems, the group profiles of users or documents are more likely to reflect the common preference or relatedness of like-minded users or documents with similar functionality. In the following section, we will discuss the group profiling approach by using clustering.

4.2 Similarity Matrixes for the Users and Documents

The relationship among all of the users is to calculate the similarity. The similarity is quantity that reflects the strength of relationship between two objects. In the last part, each user profile can be represented by the pair of tags and frequencies. We utilized the cosine distance between users. Its value ranges from 0 to 1, the higher value of the similarity, the more similar the objects are. The similarity matrix $SM(U_i, U_j)$is given by,

$$SM(U_i, U_j) = \frac{UP_i \cdot UP}{|UP_i| \times |UP_j|}$$

The users' relationship can be represented in the form of bipartite graph model. Given a graph $G = (U,E)$, where U is a set of users as $U_i = \{U_1, U_2, \cdots, U_m\}$, and E is a set of edges which entry $SM(U_i, U_j)$ reflects the similarity between users, the similarity matrix $SM(u_i, u_j)$.

Similarly with the document profiles, we can define a graph with N users $G = (D,E)$, where D is a set of documents as $D_i = \{D_1, D_2, \cdots, D_n\}$, and E is a set of edges which entry $SM(D_i, D_j)$ reflects the similarity between documents. The similarity matrix $SM(D_i, D_j)$is given by:

$$SM(D_i, D_j) = \frac{DP_i \cdot DP_j}{|DP_i| \times |DP_j|}.$$

4.3 Group Profiling via Clustering Algorithm

To accomplish the group profiling, one of the approaches is to group the user profile and document profile into several groups based on the similarity so that the objects in the same groups can share tag set. The clusters of users or documents reveal the common user preference or relatedness of documents. It can benefit the user in the same group to share the similar interests or documents.

Clustering algorithm aims to assign a set of observations into subsets so that observations in the same cluster are similar in some sense. It is specially designed for transactional data and can efficiently partition historic user transactions into clusters [8; 14; 12]. Each cluster is a set of transactions representing the interests of a particular user group. So it can find the potential groups from the user profile and document profile.

The object of clustering is: similar objects have high similarity, the similarity is low between objects of different clusters. Clustering is the process to adjust the ranks of the similarity matrix, by a number of matrix blocks to meet the similarity value larger among the inside elements, while the similarity value is small between the clusters.

There are a lot of clustering algorithms such as k-means, fuzzy c-means, single linkage and so on. In clustering analysis, almost all approaches are based on the similarity between subjects to partition the data points. Various clustering approaches have different advantages and drawbacks. Among the traditional clustering algorithms, spectral clustering has the superior capability of effectively group data by leveraging the statistical property of similarity matrix of data.

In this paper, we will introduce the Spectral Clustering Algorithm. Spectral clustering refers to a class of techniques which rely on the eigenvalues of the adjacency similarity matrix; it can partition all of the points into disjoint clusters, the points that have high similarity will be classified under the same cluster. One cluster's points have low similarity with other clusters' points. The spectral clustering is based on the graph partition. We have explained how to get the similarity matrix from the graph in 4.2. It maps the original inherent relationships onto a new spectral space, on which the user or document profile is projected. After the projection, the whole user or document profiles are simultaneously partitioned into disjoint clusters with minimum cut optimization.

Compared to those algorithms, spectral clustering has many fundamental advantages. Results obtained by spectral clustering often outperform the traditional approaches, it is easy to implement and can be solved efficiently by standard linear algebra methods. Spectral clustering techniques make use of the spectrum of the similarity matrix of the data to perform dimensionality reduction for clustering in fewer dimensions.

The original formula for the spectral clustering is:

$$L = I - D^{-1/2} S D^{-1/2}$$

According to the spectral graph theory in [4], the k singular vectors of the reformed matrix $RM_{User} = D^{-1/2} SM_{User} D^{-1/2}$ present a best approximation to the projection of user-tag vectors on the new spectral space.

And the $RM_{Document} = D^{-1/2} SM_{Document} D^{-1/2}$ presents the document-tag vector on the new spectral space.

The D_u and D_d is the diagonal matrix of user similarity matrix and documents similarity matrix, which are defined as:

$$D_u(i,i) = \sum_{j=1}^{N} SM\left(U_i, U_j\right), i = 1, \cdots, M$$

$$D_d(i,i) = \sum_{j=1}^{N} SM\left(D_i, D_j\right), i = 1, \cdots, N$$

Let's take the documents set for example.

In this case we assume that the first K singular eigenvectors represent the best approximation of original profile space. Let L_s the $m \times k$ matrix of the k singular vectors of $RM_{Document}$. As our aim is to conduct a clustering on the document profile attributes, we create a new $m \times k$ matrix RV to reflect the projection of the row and column vectors on the new spectral space in lower dimension as: $RV = [D_d^{-1/2} L_s]$.

The clustering results in the group profiling. The full steps of group profiling via clustering algorithm is summarized in the below Algorithm.

Input: The N document profile collection $DP = \{DP_i \,|i = 1, 2 \cdots N\}$, $DP_j = \{(T_1, t_1), (T_2, t_2) \cdots (T_K, t_K)\}$.

Output: A set of k clusters $DGP = \{DGP_i \,|i = 1, 2 \cdots k\}$such that the cut of k-partitioning of the bipartite graph is minimized.

1. Construct the usage similarity matrix $SM_{Document}$ from the document profile, whose element is determined by the distribution of tags of all users;
2. Calculate the diagonal matrixes D_d;
3. Form a new matrix $RM_{Document} = D^{-1/2} SM_{Document} D^{-1/2}$;
4. Perform SVD (Singular value decomposition) operation on $RM_{Document}$, and obtain k singular vectors L_s create a new projection matrix RV;
5. Execute a clustering algorithm on RV and return clusters of documents:$DGP = \{DGP_i \,|i = 1, 2 \cdots k\}$.

From the above steps, the N documents are divided into t clusters, the document group profile for each cluster is: centerline$DGP_i = \{UP_{i1}, UP_{i2}, \cdots UP_{it}\} = \{(T_1, w_{i1}), (T_2, w_{i2}) \cdots (T_i, w_{it})\}$

Where $T_s \in T, s = 1, \cdots, t$ and $(w_{i1}, w_{i2}, \cdots w_{it})$ is the centroid of the document cluster DGP_i.

Meanwhile, the selection of cluster number k is another concern in the context of clustering, which is commonly encountered. The selection of k value has a straight impact on the performance of clustering: the bigger number of k results in the over-separation of users and documents, while the smaller number of it prevents the data from being sufficiently partitioned. Thus it is necessary before performing the clustering to select an appropriate value of k to achieve a better clustering performance. In the experimental part, we will investigate the study of k selection.

In similar way, the user group profile of k users can be generated in the same way: $UGP_i = \{DP_{i1}, DP_{i2}, \cdots DP_{ij}\} = \{(T_1, x_{i1}), (T_2, x_{i2}) \cdots (T_K, x_{iK})\}$

Where $T_s \in T, s = 1, \cdots, t$ and $(x_{i1}, x_{i2}, \cdots x_{it})$ is the centroid of the user cluster UGP_i.

5 Experimental evaluations

In order to evaluate the proposed group profiling, we performed experiments on the "MovieLens" dataset. Our experiments focus on the cluster number selection; demonstration of the group profiles; and the computational cost reduction.

5.1 Dataset and Modularity Metric

As for experiment dataset, we utilize the part of the "MovieLens" data, which contains tags provided by users on movies. It includes 521 users, 1399 documents and 1956 tags. The differences of results are shown when choosing different numbers of clusters in order to get the optimal number of clusters. The data is based on the average result of executing the same experiment ten times over the same dataset.

The modularity metric is one of the standard quantitative measures for the evaluation of "goodness" of the clusters. The modularity of a particular division of a network is calculated based on the differences between the actual number of edges within a community in the division and the expected number of such edges if they were placed randomly. Good divisions, which have high values of the modularity, are those dense connections between the nodes within modules but sparse connections between different modules. It will help to evaluate the quality of the cluster; i.e. the similarity of each cluster.

After clustering, we can get several clusters. Consider a particular division of a network into k communities. We can define a $k \times k$ symmetric matrix SM whose element sm_{ij} is the fraction of all edges in the network that link vertices in community p to vertices in community q. The similarity of smC_{pq} between the two clusters C_p and C_q,is defined as,[13]

$$smC_{pq} = \frac{\sum_{c_p \in C_p} \sum_{c_p \in C_q} c_{pq}}{\sum_{c_p \in C} \sum_{c_q \in C} c_{pq}}, p, q = 1, 2 \cdots m$$

where c_{pq} is the element in the similarity matrix SM. When p=q, the smC_{pq} is the similarity between the elements inside the clusters, while $p \neq q$, the smC_{pq} is the similarity between the cluster C_p and the cluster C_q. So the condition of a high quality cluster is $\arg\max(\sum_p smC_{pp})$ and $\arg\min(\sum_{p,q} smc_{pq}), p \neq q, p, q = 1, 2, \cdots m$.

Summing over all pairs of vertices in the same group, the modularity, denoted Q, is given by:

$$Q = \sum_{p=1}^{m} [smc_{pp} - (\sum_{q=1}^{m} smc_{pq})^2] = TrSM - \left\|SM^2\right\|$$

where the m is the amount of clusters. The trace of this matrix $TrSM = \sum_{p=1}^{m} smC_{pp}$ gives the fraction of edges in the network that connect vertices in the same community Clearly a good division into communities should have a high value of this trace. If we place all vertices in a single community, the value of would get the maximal value of 1 because there's no information about community structure at all.

This quantity measures the fraction of the edges in the network that connects vertices of the same type minus the expected value of the same quantity in a

network with the same community divisions. Utilizing the value Q to evaluate the clusters [13] is a common method: the values approaching Q=1, which is the maximum value, indicate the networks with strong community structure. In practice, the values of such networks typically range from 0 to 1. The higher value of Q, the better quality for the cluster corresponding to a predefined cluster number k. So examining the Q value allows us get the optimal number of clusters.

5.2 Experimental Results

Optimal Cluster Number Selection

Here we compare the result of Q values by using Spectral Clustering, Single Linkage Clustering and Random Clustering.

Of the entire 521 user profiles constructed, we employ various clustering algorithms to build up the group user profiles. We generate the cluster from 2 to 260 and utilize the modularity method to evaluate the results. We close the number until 260 because it is half of the total amount. When the number of cluster is higher than 260, the average number of members in each cluster is lower than 2, which will not provide reasonable clustering information. The results are shown in Fig3. that the value of Q for Spectral Clustering Algorithm is consistently higher than the other two algorithms. When the number is 25 the Q gets the maximal as 0.381. With the growth of the number of clusters, the value of Q is gradually decreasing to 0.19. It is concluded that, for this dataset, 25 clusters is the best choice.

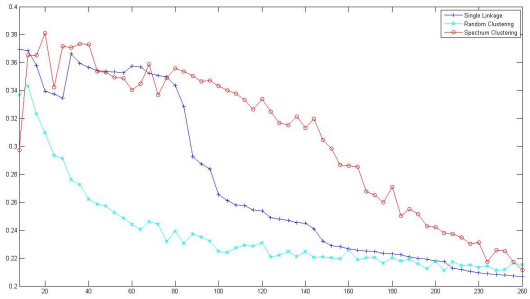


Figure 3: Comparison of the three algorithms on 521 users

Of the entire 1399 document profile models constructed, we generate the cluster from 5 to 700 and utilize the modularity method to evaluate the results. Similarly as the user profile, we close the number until 700. As shown in Fig4, when the number is 20 the Q gets to the maximum at 0.277. With the growth of number of clusters, the value of Q is decreasing to 0.026. We then found that, for this dataset, 20 clusters is the best choice.

Demonstration of the Group Profiles

It is shown that the 20 clusters is the optimal number for the 1399 documents, we take 2 clusters of them for analysis. One of the cluster contents 51 documents, with 239 tags. The main tag in it is about the "classic", "based on a book", "black and white", "National Film Registry", "breakthroughs", "Disney" and so on.

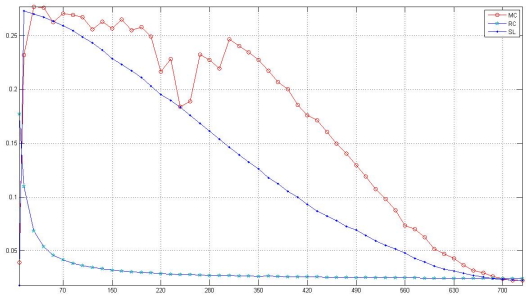


Figure 4: Comparison of the three algorithms on 1399 documents

The movies in such cluster seem related to the movies about life.

And another cluster has 79 documents with 397 tags, the dominant tags are "action", "organized crime", "guns", "hysterical", "USA film registry", "afternoon section", "Oscar (Best Actor)", "Oscar (Best Cinematography)", "Oscar (Best Director)", "Oscar (Best Picture)" and so on. Such movies tend to the Oscar movies with some breathtaking content.

Comparison between the Time Consuming

When the user or document profiles are used in tagging systems for further applications, similarity calculation is a major operation involved. An advantage of group profiling is the possibility of reducing the computational complexity. For example, cosine similarity is often executed to determine the ranking of candidates. The traditional way is to calculate the similarity between its tags and each document's tags. In such way the time complexity is the $O(n)$. It will cost much time when the system has large dataset.

After clustering for all of the documents, each cluster will have its own centroid as the representation of the group profile, which means the "center point" in the cluster. Centroid can be generated by the average frequency of tags inside the cluster. The similarity between the centroid and the items inside the same cluster should be the highest; the similarity between the centroid and the items inside the other clusters should be the lowest. So the centroid is the representative of the cluster. If N documents have clustered into m communities, the process of similarity calculation is divided into two steps: firstly, calculate the similarity between the target tag vector and the centroids of all clusters to determine the cluster with highest similarity score; secondly, calculate the similarity of the target tag vector with the document profiles inside the cluster to rank the whole documents. Since the number of centroids, m, is equal to the number of communities, which is highly lower than the number of documents, N, the time consuming of calculating similarity is dramatically reduced from $O(N)$ to $O(m+\frac{N}{m})$.

In our experiment, the time consuming computing the similarity for all 1399 documents respectively is 152.83 seconds, however, it just needs 0.045 seconds by our proposed approach to get the final ranking of documents.

6 Conclusion and future work

In this paper, we discuss an algorithm for user group profile and document group profile in social tagging systems. Utilizing the clustering algorithm, group profiling can be processed by the user profile and document profiles. We implement experiments on real tagging dataset to validate the proposed approach, investigate the modularity method to compare the optimal number of clusters, and demonstrate the content of clusters. At last, we compare the time consuming for the similarity calculation involved in real applications. It is shown that the group profiling can be dealt with the tasks outlined in the paper effectively.

For the future work, we intend to conduct research on the optimization for the algorithm, and explore the deployment of group profiling in tag-based recommender system. We will investigate clustering for tags which we believe should help in tag recommendation and representing user interests.

References

[1] F. Abel, N. Henze, D. Krause, and M. Kriesell. On the effect of group structures on ranking strategies in folksonomies. *Weaving Services and People on the World Wide Web*, pages 275–300, 2009.

[2] K. R. Bayyapu and P. Dolog. Tag and Neighbour Based Recommender System for Medical Events. In *Proceedings of MEDEX 2010: The First International Workshop on Web Science and Information Exchange in the Medical Web colocated with WWW 2010 conference*, 2010.

[3] J. Davies, D. Fensel, C. Bussler, and R. Studer. The Semantic Web: Research and Applications. In *Proceedings of the First European Semantic Web Symposium*, 2004.

[4] I. S. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 269–274. ACM, 2001.

[5] J. Diederich and T. Iofciu. Finding communities of practice from user profiles based on folksonomies. In *Proceedings of the 1st International Workshop on Building Technology Enhanced Learning solutions for Communities of Practice*. Citeseer, 2006.

[6] F. Durao and P. Dolog. A personalized tag-based recommendation in social web systems. *Adaptation and Personalization for Web 2.0*, page 40, 2009.

[7] F. Durao and P. Dolog. Social and Behavioral Aspects of a Tag-Based Recommender System. In *Ninth International Conference on Intelligent Systems Design and Applications, 2009. ISDA'09*, pages 294–299, 2009.

[8] F. Durao and P. Dolog. Extending a hybrid tag-based recommender system with personalization. In *Proceedings of the 2010 ACM Symposium on Applied Computing*, pages 1723–1727. ACM, 2010.

[9] S. Guha, R. Rastogi, and K. Shim. Rock: A robust clustering algorithm for categorical attributes* 1. *Information Systems*, 25(5):345–366, 2000.

[10] A. Hotho, R. Jäschke, C. Schmitz, and G. Stumme. Folkrank : A ranking algorithm for folksonomies. In K.-D. Althoff and M. Schaaf, editors, *LWA*, volume 1/2006 of *Hildesheimer Informatik-Berichte*, pages 111–114. University of Hildesheim, Institute of Computer Science, 2006.

[11] Q. Mei and K. Church. Entropy of search logs: how hard is search? with personalization? with backoff? In *Proceedings of the international conference on Web search and web data mining*, pages 45–54. ACM, 2008.

[12] A. Nanopoulos, H. H. Gabriel, and M. Spiliopoulou. Spectral Clustering in Social-Tagging Systems. *Web Information Systems Engineering-WISE 2009*, pages 87–100, 2009.

[13] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical review E*, 69(2):26113, 2004.

[14] M. G. Noll and C. Meinel. Web search personalization via social bookmarking and tagging. In *Proceedings of the 6th international semantic web conference and 2nd Asian conference on Asian semantic web*, pages 367–380. Springer-Verlag, 2007.

[15] S. Rendle, L. B. Marinho, A. Nanopoulos, and L. Schmidt-Thieme. Learning optimal ranking with tensor factorization for tag recommendation. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 727–736. ACM, 2009.

[16] J. Stoyanovich, S. Amer-Yahia, C. Marlow, and C. Yu. Leveraging tagging to model user interests in del. icio. us. *AAAI SIP*, 2008.

[17] P. Symeonidis, A. Nanopoulos, and Y. Manolopoulos. A unified framework for providing recommendations in social tagging systems based on ternary semantic analysis. *IEEE Transactions on Knowledge and Data Engineering*, 2009.

[18] J. Teevan, M. R. Morris, and S. Bush. Discovering and using groups to improve personalized search. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, pages 15–24. ACM, 2009.

[19] Y. Yang and N. Marques. User group profile modeling based on user transactional data for personalized systems. *Progress in Artificial Intelligence*, pages 337–347, 2005.

[20] C. M. A. Yeung, N. Gibbins, and N. Shadbolt. A study of user profile generation from folksonomies. In *Social Web and Knowledge Management, Social Web 2008 Workshop at WWW2008*. Citeseer, 2008.

Modeling, obtaining and storing data from social media tools with Artefact-Actor-Networks

Wolfgang Reinhardt, Tobias Varlemann, Matthias Moi and **Adrian Wilke**
University of Paderborn
33102 Paderborn, Germany
{wolle,tobiashv,moisun,wilke}@uni-paderborn.de

Abstract

Social interaction between people has peerlessly changed with the availability of the Internet and the World Wide Web. The Internet brought new ways of communication technologies to live and enhanced people's reachability, augmented possibilities for personal presence and the sharing of information objects. People are engaging in social networks in a steadily growing manner and share information objects within their communities. The high initial amount of data in such networks can serve as foundation of serious investigations towards social interactions of communities of learners. In this paper we introduce the technological foundation and architecture to model, obtain and store such user and object information in so-called Artefact-Actor-Networks. Artefact-Actor-Networks combine classical social networks with artefact networks that are constructed by the use of the information objects and their connections.

1 Introduction

The Internet has evolved to be the most frequently used medium of the 21st century and it is steadily growing. The so-called Web 2.0 movement [O'Reilly, 2005] engaged people in the active production of content on the Web and fostered technology-mediated human interactions in social networks. People are heavily taking part in social networks for individual learning and knowledge work as well as for leisure activities. The huge amount of data in such networks can serve as foundation of serious investigations towards social interactions in communities of learners, knowledge workers or people spending their leisure activities online.

There already is an existing body of knowledge about the properties of social networks such as Facebook (see [Ellison *et al.*, 2007]), Twitter (see [Java *et al.*, 2007; Ebner and Schiefner, 2008; Reinhardt *et al.*, 2009a]), in Social Bookmarking systems and learning object repositories [Vuorikari, 2009] but most often this work stay focused on the relational and structural part of social network analysis [Granovetter, 1983]. Nevertheless, those studies are often combined with qualitative data from interviews or online questionnaires and thus can serve as useful information source for future research. Despite their undoubted usefulness, they do often not consider the artefacts resulting from online interactions, their content, structure and relations. Furthermore, the relations between artefacts and their creators, editors or linkers are not considered for making claims about a community. In [Reinhardt *et al.*, 2009b] we introduced the model of Artefact-Actor-Networks (AANs)[1] which tries to overcome those limitations by considering both content of artefacts and the relations to actors interacting with them. As artefact we consider any digital information object that is shared within a communitiy (examples are simple HTML pages, status updates in microblogging services, entries in blogs, social bookmarks, online available documents, photos in picture sharing services and many more).

In this paper we focus on the technical side of AANs and introduce the architecture to model, obtain and store such user and object information.

2 Obtaining and modeling data from social networks

In this section we describe the concept of Artefact-Actor-Networks and the ontologies in place to model both actors in social networks and the artefacts resulting from their interaction. Moreover, we will reference existing standards and vocabularies for modeling objects and their interactions in online communities as well as metadata describing the respective properties of such objects.

2.1 Artefact-Actor-Networks

Artefact-Actor-Networks (AANs) are an approach to semantically intertwine social networks with so-called artefact networks. The theoretical model and a first implementation were first introduced by [Reinhardt *et al.*, 2009b].

To connect artefacts and actors with each other, semantic relations are required. Relations in the network are connecting objects by a semantic context like *isAuthor* or *isRightHolder*. With the help of Artefact-Actor-Networks participation in the life cycle of artefacts as well as significant connections to involved actors will be outlined. Artefact-Actor-Networks consolidate multilayered social networks and artefact networks in an integrated network. Therefore, we consider the communication and collaboration with each communication tool or artefact supply (e.g. Twitter, chats, e-mail or scientific documents) as a single layer of the respective network. We unite these single layers in both social and artefact networks to consolidated networks that contain all actors and artefacts respectively (cf. figure 1). While in the consolidated social network we can only make statements concerning the relations between actors and in the consolidated artefact network we

[1] See `http://artefact-actor-networks.net`

can only analyze the relations between artefacts, Artefact-Actor-Networks (cf. figure 2) also contain semantic relations between actors and artefacts.

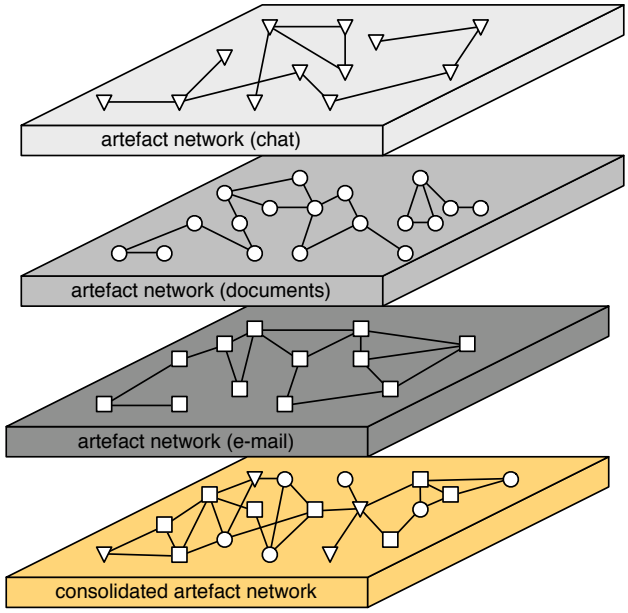


Figure 1: Consolidated artefact network resulting from three layers

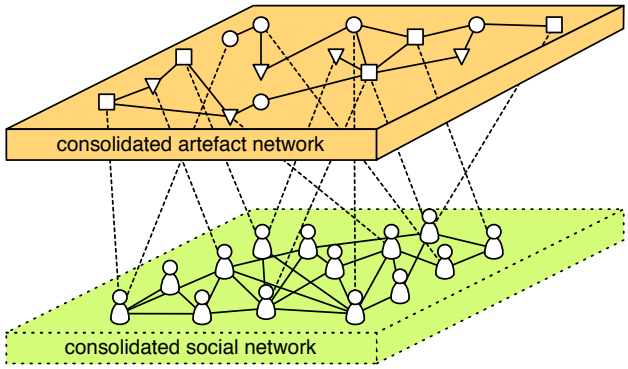


Figure 2: Artefact-Actor-Network with semantic relations between artefacts and actors

In Artefact-Actor-Networks we discern three types of semantic relations: those between artefacts (ART^2 relations), those between actors (ACT^2 relations) and finally relations that exist between actors and artefacts (AA relations). Each kind of relation can be used for certain types of analyses and supports a different type of awareness in cooperative settings. ACT^2 relations describe the nature of relationships between involved people. They characterize simple connections, friendships or kinships. Furthermore, they can show the kind of media people are communicating with. The Friend of a Friend (FOAF) project [FOAF, 2010] developed a RDF vocabulary to express interests, connections and activities of people. ART^2 relations on the other hand provide information on how artefacts are connected. The Dublin Core metadata standard and the SIOC project currently provide an expedient starting point [DCMI, 2010; SIOC, 2010]. Lastly, AA relations describe the semantics of relations between actors and the artefacts they interacted with. Dublin Core and SIOC provide useful relations to build upon, but the learning objects metadata standard (LOM) could be taken into account as well.

2.2 Relevant ontologies

During the modeling of the application domain however, we found out that we needed to extend the beforementioned ontologies[2] and vocabularies in order to cover the specifics of interaction with social media in learning networks. Thus, we created several ontologies for the social media services we are analyzing (see Section 3.3) and made our ontologies publicly available[3]. The relations build on already existing standards for the modeling and storage of metadata and are further extended by our application. Figure 3 shows a simplified overview of the ontologies used in Artefact-Actor-Networks, where AANBase defines the basic entities Actor, Artefact and Keyword. AANMeta is an ontology that allows the aggregation of multiple actors (online handles) in one real person and the relationship between so-called groups to real people, their actors and artefacts related to a group (for example artefacts that are tagged with one of the groups tags). The AANOnline ontology is used to differentiate between artefacts resulting from online actions and such artefacts that relate to activities taking place offline. The more specific tools and the respective ontologies are located towards the right of Figure 3.

3 The architecture of Artefact-Actor-Networks

The architecture of Artefact-Actor-Networks is composed of a backend to which several frontends can be connected. The backend is responsible for processing and storing data and exposing this data to the frontends by well-defined interfaces.

The architecture is based on the OSGi Service Platform, which is a component framework based on the Java platform. Due to the use of a specified component structure the backend is easily expandable and components can be deployed during runtime. The communication between the components is ensured with techniques from OSGi.

It is the backend's task to examine contents, to analyze, store and provide the annotated data. These four tasks are reflected in the according blocks in the AAN architecture (cf. Figure 4).

The crawling block loads contents and generates (according to the ontology) structured data. The datastore block stores all data generated and serves as the connection between all the existing blocks. The analyzer block contains several components with which a further refinement of the generated data is achieved. The fourth block provides the interfaces for various frontends. These blocks and the contained components are described in the following.

[2]Ontologies are a common way to model problem domains in an extensible and open format that is usable in various contexts. [Lohmann and Riechert, 2010] note the very precise and popular definition of the term ontology given by [Gruber, 1993] who notes that an ontology is "*a specification of conceptualization*" . Furthermore [Maalej *et al.*, 2008] prod to the fact that ontologies normally are valid for a much longer time than conceptual models for example, as they describe a broader application domain and some more general knowledge facts.

[3]See `http://artefact-actor-networks.net/ontologies/2010/03/`

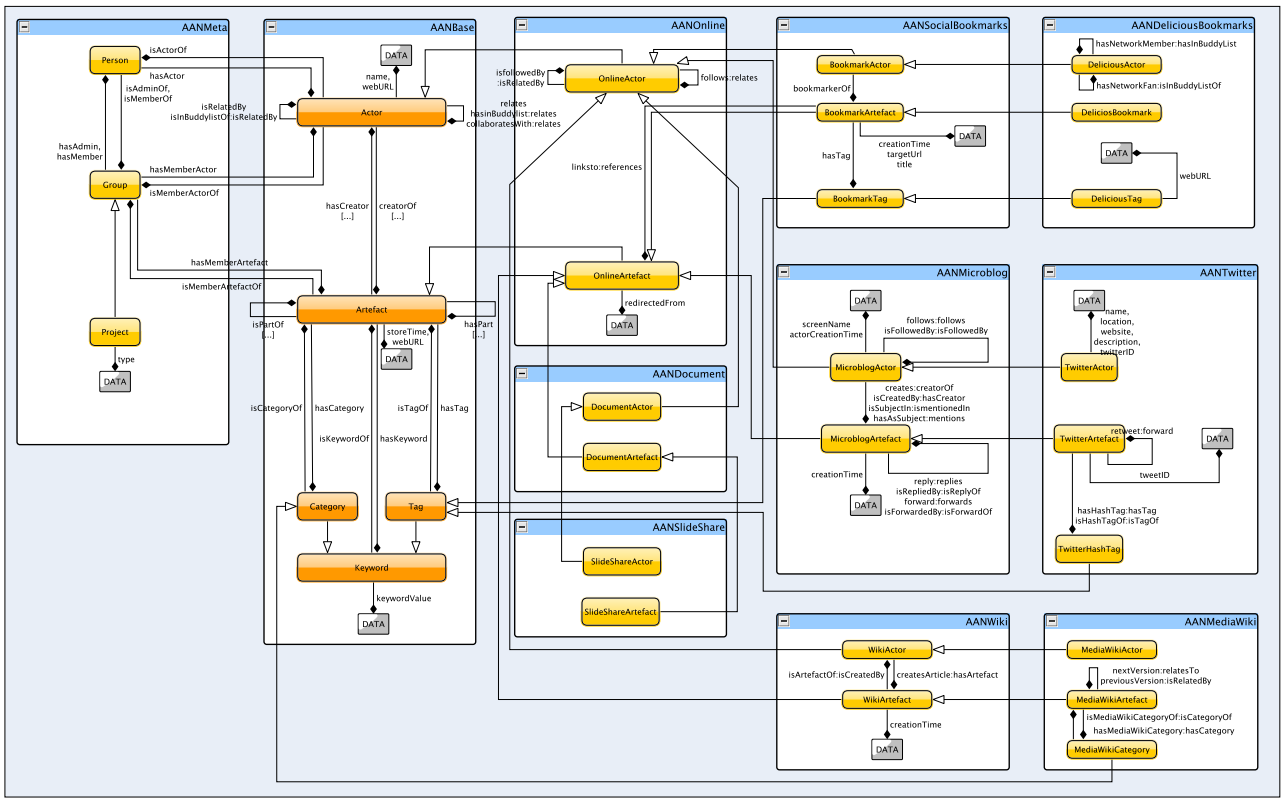


Figure 3: Simplified overview of the AAN Ontologies

3.1 Crawling-Block

At the crawling block, requests for content analysis are processed. These requests are passed to the crawling block via OSGi-Services and analyzed by a processing chain. There are two interfaces: The components Crawler and CrawlerManager. The CrawlerManager contains high-level functions by which the Crawler is controlled. The processing chain is controlled by the crawling component. It consists of the components Accessor, which is reading the content, MimeTyper, to identifier the type, and Parser, which analyzes the content.

Crawler

The Crawler component provides low-level functions to process tasks. Tasks are committed as URIs of the content. For such an URI, the Crawler executes the processing chain by loading, determining the MIME type and analyzing the content. The selection of the Accessor component depends on the used protocol and the URI. Accordingly, the MIME type is determined, what in turn selects the Parser to store contents in the datastore block. While the Parser is chosen, it is taken into consideration, if a specialized Parser exists that is able to handle variations of the detected MIME type. An example is a request of a web service, which supplies a MIME type *text/xml*. The use of a Parser that is able to handle this special type is more practicable than the use of a Parser, which can process all XML formats.

The processing of tasks occurs asynchronously by the use of a thread pool with which several tasks can be executed parallel.

CrawlerManager

A superordinate of the Crawler component is the CrawlerManager, which is using the services of the Crawler. The CrawlerManager provides functions of higher levels than the Crawler. Whereas the Crawler is receiving tasks via an exact URI, the CrawlerManager is designed to handle more complex jobs. It is possible to deal with individualities of web pages or investigate the structure of a page by following hyperlinks in HTML documents.

To handle special tasks, there can be various implementations of the CrawlerManager. Two examples, the GenericCrawlerManager and the MediaWikiCrawlerManager, can be seen in the architecture (figure 4). The GenericCrawlerManager is able to process timed tasks, which can generate follow-up tasks. The MediaWikiCrawlerManager is specialized to the structure of MediaWiki pages and is able to crawl contents of an entire wiki.

Accessor

The first element of the processing chain is the Accessor component. It is responsible for reading content. The component is responding to the Crawler, which hands over an URI of the content to load. The Accessor component accesses the resource, stores it in a local file, and returns a reference to the file.

Different types of the Accessor component are conceivable. Those could enable the access to protocols like HTTP, FTP, SMTP or SVN.

MimeTyper

For further processing of a resource by a Parser the MIME type of resource is determined. This improves the choice of an appropriate Parser in the last stage of the processing chain. The MimeTyper is loading the temporary, local file of the content to analyze. Then it determines one or more MIME types of the content and returns it to the Crawler.

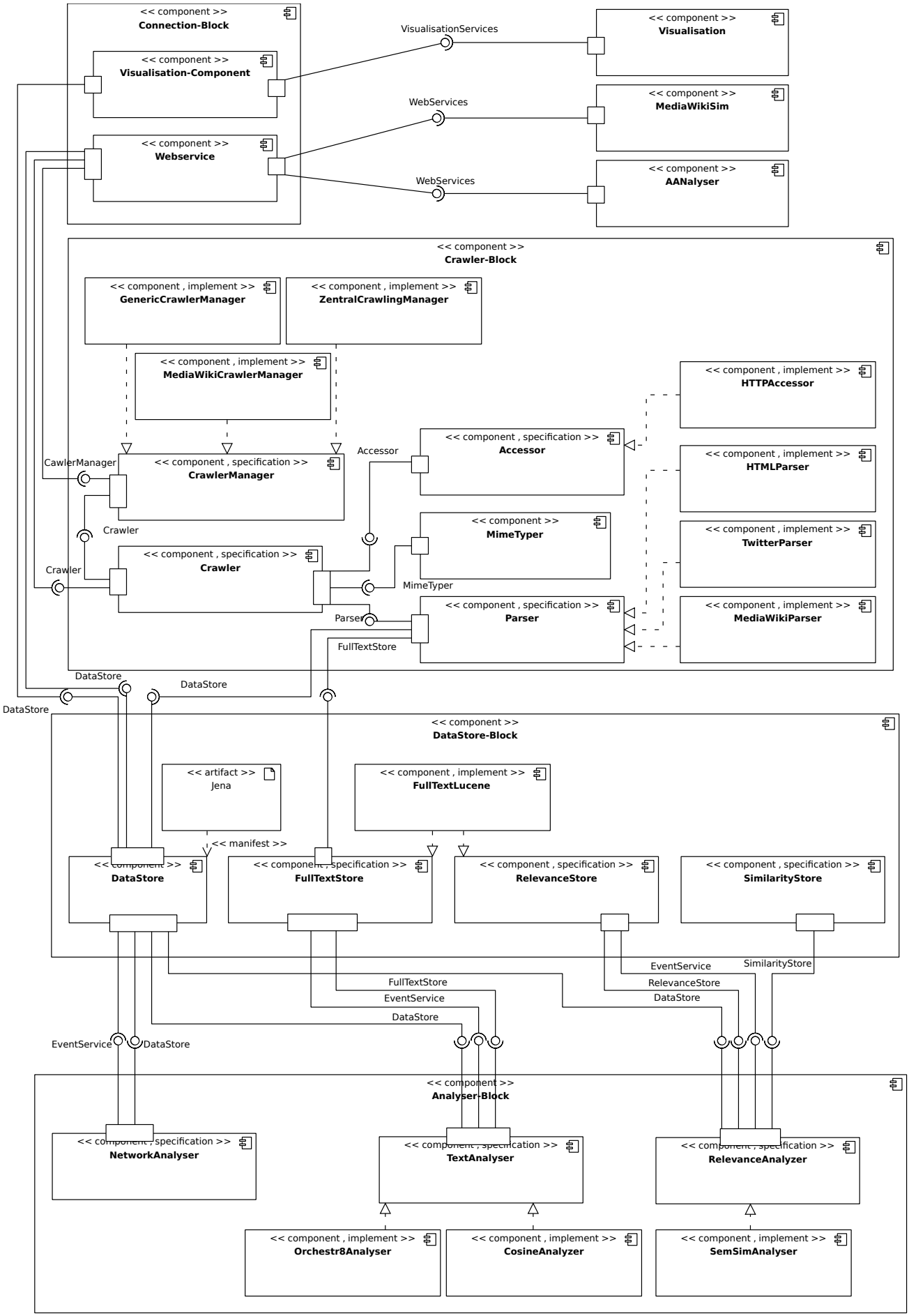


Figure 4: The AAN architecture

Parser
The Parser is the last stage of the processing chain. It extracts relevant information from the resources loaded by the Accessor, to build up an Artefact-Actor-Network. Different variations of Parsers are distinguished by the Parser type and the MIME types which can be processed. Types are setting in advance the order, in which the Parsers are chosen. Parsers which can handle a specialized MIME type are preferred to general Parsers. This guarantees the choice of a Parser which can process contents optimally.

3.2 DataStore block

The DataStore block is responsible for the storage of all data produced in the application. For this purpose, four components are available to accommodate the differently structured data. Other components will be notified of any changes via an event service. All components in the DataStore block send custom events for communication purposes.

DataStore
The Parsers generate RDF data from the analyzed contents that is stored in the DataStore according to the respective ontology. RDF data provides several advantages for Artefact-Actor-Networks. In addition to the flexibility of the data structure and the ability to generate descriptions of the data at runtime, there is a wide range of storage engines that are adapted to the processing of RDF data. The backend uses the Jena Framework[4] to store and query the data. With Jena data can be queried both in source code and via the RDF data query language SPARQL[5]. The DataStore stores the data in a file format defined by the Jena Framework but one can also use in-memory storage or a RDBMS.

FullTextStore
Some analytical techniques for text-based content – such as keywords, named entities or related languages – require the existence of a full text. The texts can be extracted by a Parser and will be stored in the FullTextStore. The FullTextStore will notify components in the analysis block about the existence of a full text that can be retrieved via an event system.

RelevanceStore
The RelevanceStore holds information about the relevance of a keyword. This relevances are used during the calculation of semantic similarity of artefacts. In our architecture the FullTextStore and the RelevanceStore were incorporated in one component. We use the Apache Lucene engine to efficiently store the full texts and to query the FullTextStore for the relevance of single keywords.

SimilarityStore
The SimilarityStore manages all calculated similarity values between artefacts, actors, persons and groups. Similarity values are created by various similarity analyzers, which store their results in the SimilarityStore according to an analyzer-specific order.

3.3 Analysis block

The Analysis block includes all components for analyzing stored data of components from the DataStore block. Components in this block only listen to events fired from DataStore components. We distinguish between NetworkAnalysers, TextAnalysers and RelevanceAnalysers. NetworkAnalysers listen to events fired from the DataStore, Textanalysers listen to events from the FulltextStore and RelevanceAnalysers to events from the RelevanceStore.

NetworkAnalyser
NetworkAnalysers are using the data pool of the DataStore to analyse network structure. They will be informed to every change in the data pool of the DataStore.

TextAnalyser
TextAnalysers start working upon the reception of FullTextStore events. Currently we implemented Orchestr8Analyser, OpenCalaisAnalyser and CosineAnalyser. The Orchestr8Analyser and OpenCalaisAnalyser are making use of the webservices of Orchestr8 and OpenCalais respectively to extract keywords and named entities from a given text. Extracted data will be stored as additional parts of resources.

RelevanceAnalyser
RelevanceAnalysers are components to determine the similarity of resources. An example for such an algorithm could be the SemSim algorithm that we developed (see [Reinhardt *et al.*, 2009b] for more detail on the algorithm). SemSim allows the calculation of semantical similarity based on the keywords and named entities stored for artefacts. The calculated values are then stored in the SimilarityStore.

3.4 Connection block

The fourth block in our architecture is the Connection block that encapsulates Webservices and the visualization component. Both components expose data stored in Artefact-Actor-Networks or API functionalities to external consumers over HTTP interfaces.

Webservices
Internal server components such as the CrawlerManager, the CrawlerManager or the DataStore are providing their functionalities via webservice interfaces as an API to external applications. Using the API, crawling jobs can be placed or data from the DataStore can be requested. The DataStore provides tailored API functions as well as a SPARQL interface to the AAN.

Visualization component
The visualization component converts the contents of Artefact-Actor-Networks into a XML-based graphics format that can be displayed and explored in the appropriate viewers. The preparation of those files is costly and will be a great burden for the AAN server. In order to reduce the efforts we implement a caching strategy that will only keep those projects up-to-date that are being accessed regularly.

3.5 Technical note

The architecture is based on the OSGi service platform[6] a component framework based on the Java platform. The OSGi service platform is a specification of the OSGi Alliance, which has been implemented in various implementations. Amongst those implementations are Eclipse Equinox[7] and Apache Felix[8]. The frameworks support the live deployment of components, which allows to add a new Parser without the need to restart the server.

[4] http://openjena.org/
[5] http://www.w3.org/TR/rdf-sparql-query/
[6] http://www.osgi.org
[7] http://www.eclipse.org/equinox
[8] http://felix.apache.org

OSGi also allows the dynamic communication of components with each other. For this purpose, a service system is used that allows components to register services that then can be used by other components. The AAN architecture makes heavy use of this approach, as all components provide specialized services and await use. As an example, each Parser has to provide a service that can be accessed via the two methods *isParsable* and *parse* of the Crawler component.

4 Social media tools under investigation

In its current implementation of AANs we store and analyze data from four different social media tools: (1) Twitter, (2) Delicious, (3) SlideShare and (4) MediaWiki. All of the tools are used by researchers during their daily work routines (see for example [Heinze *et al.*, 2010] for an inspection of tools used by researchers) and make specific demands on the respective components in the AAN architecture. In the following we present the specifics of the single components.

4.1 The specifics of the Twitter component

Twitter[9] is a microblogging service, which allow users to publish short messages with an length under 140 characters. These messages are typically public and can be viewed over different channels. The *TwitterParser* component use the answers of the TwitterAPI in XML or JSON format to parse the information of a single Tweet(Status), a Twitteruser, a users timeline, the followers of a User or a search request. The *TwitterParser* possesses a special component for each of this functions which will be described now.

The *StatusComponent* parses the XML answer of the TwitterAPI */show/status* request. It extracts the information about the status and the user who created it. The extracted information is kept in the *DataStore* as listed below:

CreationTime the creation time of the status.

Statusid the Twitter id of the status.

Replyid the Twitter id of the status to which the current status is the reply.

User the information of the creator of this status. This will use the *UserComponent*.

WebURL the URL to this status as an HTML page.

Hashtags the hashtags of this status.

ExternalLinks hyperlinks that this status may contain.

Text the full text of the status that will be stored in the *FullTextStore*.

TwitterAPI */show/user* requests are processed in the *UserComponent* which extracts information about the Twitter user and stores them in the *DataStore*. The extracted information are listed below:

UserID the Twitter id of the user.

Screenname the screen name of the user.

Username the real name of the user.

Location the location where the user lives.

Description the description of the user which was entered in Twitter.

URL the internet address of the user.

CreationTime the date at which the user registered at Twitter.

Last Status the last status of the user (will passed to the *StatusComponent* for analysis).

The TwitterAPI response for a timeline contains a series of statuses. This series will be separated by the *TimelineComponent* into the single statuses. At the last step the *TimelineComponent* will forward each single status to the *StatusComponent* which will extract the information.

The */show/followersid* API call responds with a list of TwitteruserIDs which will be parsed and returned to the *Crawler* as follow-up links, which can be followed by a *CrawlerManager* to parse the user information of the followers.

The response from the TwitterSearchAPI is computed the same way as followers. The StatusID will be extracted from the response and returned to the *Crawler* as follow-up links. This is necessary because Twitter uses a different data structure in the SearchAPI as in the rest of the TwitterAPI.

4.2 The specifics of the Delicious component

One of the integrated data sources is the social bookmarking service Delicious[10]. Delicious can be used to store personal bookmarks on the web and share them with others. During the creation of bookmarks users have the opportunity to add notes and tags to describe and categorize their input. By adding this additional data, especially the tags, artefact-networks are created. On the one hand bookmarks of different users form networks by relations resulting from their tags. On the other hand all bookmarks of a user are connected to the user himself. Besides these artefact-networks, actor-networks can also be found at Delicious. These are formed while users add other Delicious users to their personal network. By these relations, some users are connected indirectly, as well as their bookmarks are connected additionally. In summary, Delicious provides both types of networks that form the base of Artefact-Actor-Networks. What is the most practicable way of proceeding to get it?

Data access

Delicious offers a huge amount of possibilities for developers to access the available data[11]. Depending on the desired outcome, one can choose from different interfaces, e.g. an API, feeds or link-rolls. Generally, one of the most applied ways to access data is by the use of an API. We also tested this way for applicability to our system. The offered API-methods are custom-made for the access and use of a users personal data. A user can create, edit and receive personal bookmarks, tags and tag bundles. As the idea of AAN is the extraction and analysis of public data, and the need of a user-authentication by the API is a hindrance, there was a need for more useful data access.

A more utilizable approach to get data is the use of Delicious feeds[12]. Feeds are offered in JSON and RSS format and provide an access to public data. It is possible to get the latest bookmarks, tags and network members of a specified user. Furthermore, requests for bookmarks can be refined by combining a specific username and tags. Moreover, recent bookmarks for an URL can also be accessed what forms an extensive base for information retrieval.

[9] http://www.twitter.com/

[10] http://delicious.com/

[11] http://delicious.com/help/tools

[12] http://delicious.com/help/feeds

As feeds are mainly used for receiving the latest information, most of the feeds are provided as a list of recent bookmarks. Furthermore, the Delicious feeds are limited to 100 entries per request and as there is also a limitation of one request per second we encountered a restriction for crawling the entire bookmarks of a user. This fact was partly solved by multiple recursive requests. If a user has described his bookmarks with the tags A, B and C, first the bookmarks described with tag A are requested. If the returned set of bookmarks amounts to 100 entires, a combined request of tag A and B is sent. If the result of this request amounts to 100 entries again, a refinement by an additional tag is used. Otherwise, the bookmarks of the tags A and C are requested to get a result that is as complete as possible.

Within our system, the *DeliciousParser* is setting properties for contents of specific feed calls. With these properties, extracted tags and the count of returned artefact entries are stored. This forms the basis, by which the *DeliciousCrawlerManager* is generating feed URLs to follow up crawling a complete set of bookmarks.

Finally, the received feed-data is mapped to the defined ontology and added to the AAN model.

4.3 The specifics of the SlideShare component

SlideShare[13] is a Web 2.0 platform which offers users the possibility to share presentations and documents. A user can upload files in PDF and common office formats and is able to define metadata like tags, category and visibility information. Published slides can be favored, rated, commented on, downloaded and shared with others.

For developers, SlideShare provides an API[14] with which public and private data can be accessed. Public API methods require an optional user authorization. By using public methods, developers can request documents related to users or tags. Additionally, a user's tags or contacts can be requested as well.

For accessing data of the SlideShare network, the expandability of the AAN framework is used. Here we took advantage of the clear URL scheme of the API methods and thus the existing components CrawlerManager and ZentralCrawlerManager work together with the SlideShareParser, a specialized parser to analyze the incoming SlideShare data. New crawling tasks are added to one of the CrawlerManagers. If such a task consists of a SlideShare URI, the specialized SlideShareParser determines that it is able to handle the given input. If the parser is chosen it firstly analyzes the URI scheme. In the following, artefacts, actors and metadata are extracted and stored accordingly to the specified ontology. Finally, API URLs of related actors, artefacts and keywords are generated and added to the crawling queue.

4.4 The specifics of the MediaWiki component

The MediaWiki component was designed to receive as much information as possible from a MediaWiki installations such as WikiPedia. First we designed a specialized MediaWikiCrawlerManager, which is able to control the crawling process to crawl and observe a complete MediaWiki or just a single page. Furthermore, we implemented the MediaWikiParser, who's task it is to parse input that was received by the Crawler (see Section 3.1). The MediaWikiParser stores extracted information like internal and external links or information about the author within the DataStore component. The full text will be stored with the FullTextStore component. We distinguish between three different types of jobs handled by the MediaWikiCrawlerManager, which will be discussed below.

Crawling single pages

In this case, the MediaWikiCrawlerManager handles the job in four steps. First a unique URL as a permalink[15] will be generated. As discussed before every object, like a MediaWiki article is represented as a unique artefact in the Artefact-Actor-Network. Secondly the MediaWikiCrawlerManager generates an MediaWiki API-Query of the type 'parse'. The required information is initially received from the MediaWiki server of the article. Note, that this the query is not executed by the MediaWikiCrawlerManager but only the appropriate URL will be created. In the third step the MediaWikiCrawlerManager calls the Crawler to add a new crawl task. The resource will be fetched by one of the accessor components. If the MediaWikiParser is registered and started, the MediaWikiParser parses the resource because of the distinction between special and general parsers.

Further Properties By adding a new job to crawl only a single page one is able to define the properties to resolve internal and external links by specifying a depth value. If a the depth for internal link is 1 for example, the MediaWikiCrawlerManager will add new jobs for all received internal links.

Crawling a full MediaWiki

The MediaWikiCrawler can crawl a complete MediaWiki with all its articles in the latest revision or with all its articles with a specifiable number of revisions. By adding a job to crawl a complete MediaWiki, the MediaWikiCrawlerManager executes an API query to receive a list of all articles, including the latest revision information about an article. If one wants to get more then the current revision of an article, the MediaWikiCrawlerManager executes queries to get basically needed information about each revision. With this information, the MediaWikiCrawlerManager finally generates MediaWiki API queries of the type parse to add new single page jobs. Each of the generated jobs will be handled like described in the section about crawling single pages. Another important note is the limitation on the MediaWiki API. Only 500 article or revision entries can be received with one query. This is solved by executing serial queries.

Further properties For this job type the properties revision count and the depth of external links can be specified. The property revision count can be a positive integer values or -1 to crawl all revisions of all pages. The parameter about the depth of the external links is the same as described in the section about crawling single pages. It will be propagated to each created single page job.

Observing a MediaWiki

To keep AAN data up to date, the MediaWikiCrawlerManager supports the observation of a MediaWiki. After crawling a full MediaWiki initially, it regularly checks the MediaWiki about changes. This means that you must not always

[13] http://www.slideshare.net/

[14] http://www.slideshare.net/developers/documentation/

[15] Permalink is the unique URL of a MediaWiki page; e.g. `http://en.wikipedia.org/w/index.php?title=Java&oldid=366545644`

crawl and parse the complete MediaWiki, which would consume to much time. Only changes since the last successful crawling will be considered in a new crawling job.

General handling of jobs
All generated jobs will be stored in a threaded queue by scheduling first-come-first-serve. Each job is represented as a single thread, which allows to handle more than one job in parallel.

5 Outlook and further R&D opportunities

Artefact-Actor-Networks are analyzing interactions of learners with artefacts that are used for individual and organisational learning. The semantically enriched data is then exposes via an open API to be included in various user interfaces.

In [Reinhardt, 2010] we introduce the AANalyzer as the first awareness dashboard that build on the AAN model and will be applied to several learning communities in the course of the year 2010, which will help us to gain user feedback on the awareness support the tool offers. At the same time we are extending both the number of social media tools available for analysis in AANs as well as the quality of the AAN backend implementation. First functional tests with more than 400.000 nodes in an Artefact-Actor-Network revealed the need for improvements regarding the inferring of semantical data stored. Besides the long runtimes to calculate the inferred models, any changes in the data model require a rebuilding of the inferred model. Furthermore, the calculation of semantic similarity between artefacts, actors and groups in AANs is a challenging endeavor to overcome. At the moment the calculations are done in an online algorithm whose runtime is exponential to the number of artefacts in the DataStore. We strive for implementing an offline algorithm that makes use of caching strategies and a the data in the SimilarityStore.

Regarding the variety of awareness widgets for the users of the AANalyzer, we will extend the choice with statistics widgets and an advanced word cloud implementation that will allow for the visualisation of timely changes in the importance and use of certain terms.

References

[DCMI, 2010] DCMI. Dublin Core Metadata Initiative. `http://dublincore.org/`, 2010.

[Ebner and Schiefner, 2008] Martin Ebner and Mandy Schiefner. Microblogging - more than fun? In *Proceedings of the IADIS Mobile Learning Conference*, pages 155–159, 2008.

[Ellison *et al.*, 2007] N.B. Ellison, C. Steinfield, and C. Lampe. The benefits of Facebook" friends:" social capital and college students' use of online social network sites. *Journal of Computer Mediated Communication (Electronic Edition)*, 12(4):1143, 2007.

[FOAF, 2010] FOAF. The Friend of a Friend (FOAF) project. `http://www.foaf-project.org/`, 2010.

[Granovetter, 1983] M. Granovetter. The strength of weak ties: A network theory revisited. *Sociological theory*, 1:201–233, 1983.

[Gruber, 1993] T.R. Gruber. A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition*, 5(2):199–220, 1993.

[Heinze *et al.*, 2010] N. Heinze, P. Bauer, U. Hofmann, and J. Ehle. Kollaboration und Kooperation in verteilten Forschungsnetzwerken durch Web-basierte Medien – Web 2.0 Tools in der Wissenschaft. In *Forthcoming Proceedings of the GMW 2010 conference*, 2010.

[Java *et al.*, 2007] Akshay Java, Xiaodan Song, Tim Finin, and Belle Tseng. Why we twitter: Understanding microblogging usage and communities. In *Procedings of the Joint 9th WEBKDD and 1st SNA-KDD Workshop 2007*, August 2007.

[Lohmann and Riechert, 2010] S. Lohmann and T. Riechert. Adding Semantics to Social Software Engineering: (Re-)Using Ontologies in a Community-oriented Requirements Engineering Environment. In *Workshop-Proceedings of Software Engineering 2010*, pages 485–494, 2010.

[Maalej *et al.*, 2008] W. Maalej, D. Panagiotou, and H.-J. Happel. Towards Effective Management of Software Knowledge Exploiting the Semantic Web Paradigm. In *Proceedings of Software Engineering 2008*, pages 183–197, 2008.

[O'Reilly, 2005] T. O'Reilly. What is Web 2.0 – Design Patterns and Business Models for the Next Generation of Software. `http://oreilly.com/pub/a/web2/archive/what-is-web-20.html`, September 2005.

[Reinhardt *et al.*, 2009a] Wolfgang Reinhardt, Martin Ebner, Guenter Beham, and Cristina Costa. How people are using Twitter during conferences. In V. Hornung-Prähauser and M. Luckmann, editors, *Creativity and Innovation Competencies on the Web. Proceedings of the 5th EduMedia 2009, Salzburg*, pages 145–156, 2009.

[Reinhardt *et al.*, 2009b] Wolfgang Reinhardt, Matthias Moi, and Tobias Varlemann. Artefact-Actor-Networks as tie between social networks and artefact networks. In *Proceedings of the 5th International ICST Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom 2009)*, November 2009.

[Reinhardt, 2010] Wolfgang Reinhardt. A widget-based dashboard approach for awareness and reflection in online learning communities based on Artefact-Actor-Networks. In *Forthcoming Proceedings of the First PLE Conference 2010*, 2010.

[SIOC, 2010] SIOC. The semantically-interlinked online communities (sioc) project. `http://sioc-project.org/`, 2010.

[Vuorikari, 2009] Riina Vuorikari. *Tags and self-organisation: a metadata ecology for learning resources in a multilingual context*. PhD thesis, Open University of the Netherlands, 2009.

Meta-rules: Improving Adaptation in Recommendation Systems

Vicente Arturo Romero Zaldivar[1] and Daniel Burgos[1,2]
[1]Atos Origin SAE, Albarracin 25, Madrid 28037, Spain
{vicente.romero, daniel.burgos}@atosresearch.eu
www.atosresearch.eu
[2]International University of La Rioja
Gran Via Rey Juan Carlos I 41, 26002 Logroño, La Rioja, Spain
www.unir.net

Abstract

Recommendation Systems are central in current applications to help the user find useful information spread in large amounts of post, videos or social networks. Most Recommendation Systems are more effective when huge amounts of user data are available in order to calculate similarities between users. Educational applications are not popular enough in order to generate large amount of data. In this context, rule-based Recommendation Systems are a better solution. Rules are in most cases written *a priori* by domain experts; they can offer good recommendations with even no application of usage information. However large rule-sets are hard to maintain, reengineer and adapt to user goals and preferences. Meta-rules, rules that generate rules, can generalize a rule-set providing bases for adaptation, reengineering and on the fly generation. In this paper, the authors expose the benefits of meta-rules implemented as part of a meta-rule based Recommendation System. This is an effective solution to provide a personalized recommendation to the learner, and constitutes a new approach in rule-based Recommendation Systems.

1 Introduction

Nowadays, when the amount of information is becoming over-exceeding, Recommendation Systems emerge as the solution to find the small piece of gold in mountains of garbage. In electronic commerce, knowledge management systems, social networks, and other fields and markets, they help users find useful products, lessons or contributions. There are many inputs which can be used as information sources like i.e. user similarities with other users, user profile, and user preferences. All these inputs provide the system with valuable data to suggest the user the best way to follow or the most appropriate choice. Furthermore, people ratings [Rocchio, 1971] are another important source of information for Recommendation Systems.

Other sources of information are i.e. user interests, goals, and objectives, all of them more useful for educational applications. However, educational applications lack of enough amounts of data to establish user similarities in a precise way. In this case, recommendations are based on information stored in a user model which is extended explicitly or implicitly. There are also hybrid approaches which ask some minimum information to the user and the rest is obtained in an implicit way.

For educational applications ***rule-based Recommendation Systems*** have proved as more useful than other systems [Abel *et al.*, 2008]. In general acceptable recommendations can be obtained with a small amount of information. However, when the system achieves a better knowledge of the user, recommendations increase precision since rules evolve in parallel or new ones are included to the rule-set. In general expressing user preferences, goals and interest with rules can be difficult [Anderson *et al.*, 2003] to solve this problem complex and large rule-sets are generated. This solution carries another problem: the size and complexity of the rule-set can be unaffordable. In addition, it would be desirable to generate rules based on data extracted from a database or from the user, increasing this way user adaptability. Such generation could also be on the fly, allowing the rule-set to be up to date.

In this paper we propose a solution to this problem by the introduction of a new abstraction level: ***meta-rules***, which provide foundations for effective adaptive and personalized processes, such as in, e.g. learning. This approach has been implemented in the context of Meta-Mender, our meta-rule-based Recommendation System. In this paper, we also describe the Meta-Mender architecture and implementation to contextualize the meta-rules approach.

2 Background and Related Work

In the field of educational recommendations there are approaches like [El Helou *et al*., 2009] which use a modified page ranking algorithm for the generation of recommendations. The algorithm considers actors, activities and resources as main entities. Here user' activity is used to create a directed graph representing the entities and links between them are generated. Later a rank is assigned to nodes and this information is used to generate recommendations for a user query. This work is valuable for us because in general meta-rules use as input the user's activity for the automatic generation of rules. This input comes as in the referenced paper from actors, activities and resources. The main difference is that in our approach it will be generated a set of rules instead of a directed graph.

In the field of rule-based recommenders there are some relevant reports in the literature [Abel *et al.*, 2008], for example, describes a rule based Recommendation System for online discussion forums for the educational online board Comtella-D. Actually the system is able to call several encapsulated recommenders, collaborative filtering or

content-based recommenders, and the rules decide according to the amount and type of user data which recommender should be called. In doing so, the rules define a meta-recommender which is very interesting but tangential to this work.

An advantage of rule-based recommenders against other approaches is how easy it is to generate explanations for these systems. In many cases, it is almost impossible to explain to the user how a Recommendation System has derived a conclusion. If the user is not sure of a given recommendation and-or prefers to receive some logic-supported arguments which help him to choose a given solution, rules-based systems are the best ones prepared to solve this issue. This problem is usual in automated collaborative filtering systems [Herlocker, 1999], where the lack of explanations decreases the system acceptance and affects user trust.

RACOFI [Anderson *et al.*, 2003] is defined by its authors as a rule-applying collaborative filtering system. This system is a hybrid conformed by a collaborative filtering recommender plus a rule-based recommender. It was designed for recommending Canadian music but its authors argue that the system is content independent. RACOFI uses rules to modify, for example, ratings of items based on item similarity. This means that if a user rates an album as highly original, other albums' originality of the same author will be incremented. This paper is very useful for us because it contains a large set or rules which will be used to prove the synthesis power of a meta-rule approach.

With regards to meta-rules, we have not found anything similar to our proposal. Initially used by LISP [McCarthy *et al.*, 1985] and other languages, the closest concept is rule templates, followed by Open Rules[1] and the Object Oriented RuleML[2] approach of handling rules as data, thus generating entire rules from its component parts. These approaches are very valuable. However, our approach of producing rules using imperative programming comprises these two approaches and, at the same time, it seems to be more powerful. For instance, Meta-Mender can easily generate meta-rules, or other supplementary features, easily. In addition, as it will be shown in the next sections, using meta-rules in Rule-based Recommender Systems is not common, and it provides other benefits like maintainability, reengineering and on-the-fly generation. Finally, the introduction of another level of abstraction is very valuable for adaptation and performance.

For the implementation of a rule-based recommender using a rule-management system can be of great help. In this respect, Meta-Mender makes use of DROOLS [DROOLS] as rule engine. DROOLS is a business rule management system (BRMS) with a forward chaining inference-based rules engine (the so-called rule system). This system makes use of an enhanced implementation of the Rete algorithm [Forgy, 1982]. DROOLS is designed to allow plug-able language implementations. Currently, rules can be written in Java[3], MVEL[4], Python[5], and Groovy[6]. It is also possible to write functions to be executed as the consequence of any rule; this feature has been used to generate rules from meta-rules. It is also possible to assign a priority to rules, which is a way to address the execution order by the rule designer.

[1] http://openrules.com/index.htm
[2] http://ruleml.org/indoo/indoo.html
[3] http://www.sun.com/java/
[4] http://mvel.codehaus.org/
[5] http://www.python.org/
[6] http://groovy.codehaus.org/

3 The Meta-Mender Architecture and Implementation

As aforementioned, the Meta-Mender Recommendation System uses DROOLS as the rule engine. This rule system works as follows, first it is required to feed the engine with a set of rules (or meta-rules in this case); these meta-rules are defined by the professor or by the technical team using the application requirements. To define meta-rules is a more complex task than to define rules. In this respect, future research will be done in order to generate meta-rules automatically. A meta-rule example will be shown in the next subsection. Later some facts should be added to the engine. These facts are extracted commonly from a database, be it relational or ontological. As the facts are inserted, rules antecedents are checked for completeness and once the engine is started, rules that fulfill its antecedents are fired and the corresponding consequences are executed. The order of execution is arbitrary, so rule priority must be stated if the order of execution is important for the final result. In our case the order is important because the output of this iteration will be a file of rules and these files have a structure, so the rules that generate the header must be executed before the rules that generate the body. The output is used for a second iteration from which the final recommendations are obtained. See Figure 1 for a simplified representation of the recommender architecture and Figure 2 for an example of a meta-rules file, this file can generate rules for recommending the next course to follow in a .LRN educational application. See that at the consequence of the meta-rules a function is called, this function receives some information that allows it to write the rules wanted.

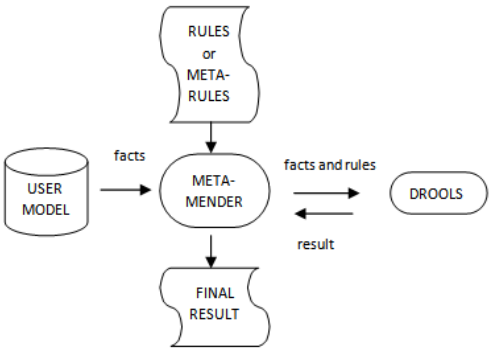


Figure 1. The Meta-Mender Architecture

It is worth mentioning that in DROOLS, rule conditions cannot contain functions, this is a common issue in rule systems the behind reason is performance. The condition is formed by a conjunction of patterns that must be matched by the inference algorithm. These patterns correspond at the end with object instances. So rule conditions are in a way fixed and must be defined statically. The only way to adapt a rule-set to changing conditions is by generating rules dynamically. These changing conditions can be:

- Changes in the application architecture, this includes the addition of a new forum or social network.

- Changes in membership conditions, access restrictions, or similar.
- Changes in model, new classes addition, modification of existing classes, etc.

It is clear that a static defined condition cannot cope with all these changes. So if there is a way for generating rules according to current system conditions and properties, it could save time and effort as long as increase adaptation. See Figure 3 for an example of a rules file obtained from the meta-rules file of Figure 2. In this figure it can be noted that the recommendation list is created as rules consequences are executed. The function `AddRecommendedCourse()` builds this list and the parameter `(10 - $courseLevel)` allows giving more priority to those courses with a lesser level of complexity. This criterion can be personalized to every student.

Meta-rules can handle changing conditions. Different rules can be generated depending on user data, system properties, etc.; also rules priority can be different which implies different recommendations.

```
package service

//global variables definitions
global java.lang.String filePath;

//rules
rule Header
//empty antecedent, executes always
  when
  then
    WriteHeader(filePath);
end

rule CourseSequenceRule
  salience -1 //priority
  when //antecedent
  //classId is an object field
    LRNClassData($classId : classId,
        $className : className)
    LRNStudentData($studentId :
      studentId,
      $studentName : studentName)
    LRNClassPerStudent(classId ==
      $classId, studentId == $studentId)
  then
    WriteClassMembershipRule(filePath,
      $studentName, $className);
end

//function implementation
function void WriteHeader(
       String filePath) {
 ...
}

function void WriteClassMembershipRule(
       String filePath,
       String studentName,
       String className) {
    ...
}
```

Figure 2. A Meta-rule File

Rule output can be related with rule priority since it will be possible to order the results and give the user a list of recommendations ordered by priorities.

4 Practical Implementations of the Meta-Mender Recommendation System

The Meta-Mender recommender is been used at present as part of two projects. The first one, TELMA[7], is focused on the application of Communications and Information Technologies in lifelong training of surgeons of Minimally Invasive Surgery (MIS).

TELMA develops an online learning environment which manages the content and knowledge generated by users in an efficient way. TELMA creates a new training strategy based on knowledge management, cooperative work and communications and information technologies aiming to the improvement of the formation process of the surgeons of MIS.

In order to cover all these needs and achieve its goals, TELMA develops a cooperative and adaptable learning platform. This platform is composed of a training system which integrates a tool for the authoring of didactic multimedia content, a Recommendation System and a social network. The Recommendation System, Meta-Mender, manages the knowledge generated by the users creating the foundations for adaptive learning. The learning platform allows the construction of a complete knowledge base thanks to the reuse and sharing of the knowledge generated for the professionals who access the learning system.

```
package service

global domain.RulesOutData outData;

rule MathI
  when
    UserData($userName : name,
      $userId : id)
    Course($courseLevel : courseLevel,
      $courseName : name, id == 1)
    not (TotalCoursePercentage(userId ==
      $userId, courseId == 1))
  then
    outData.AddRecommendedCourse(
     $courseName, 10 - $courseLevel,
     $userName);
end

rule AlgebraI
  when
    UserData($userName : name,
      $userId : id)
    Course($courseLevel : courseLevel,
      $courseName : name, id == 2)
    not (TotalCoursePercentage(
      userId == $userId, courseId == 2))
    exists (TotalCoursePercentage(
      userId == $userId, courseId == 1))
  then
    outData.AddRecommendedCourse(
      $courseName, 10 - $courseLevel,
      $userName);
end
```

Figure 3. A Rules File Obtained from the Meta-rules File of Figure 2

The second project, GAME·TEL[8], is focused on the creation of a system for the design, development, execution and evaluation of educational games and simulations,

[7] www.ines.org.es/telma
[8] www.ines.org.es/gametel

adapted to student preferences, educational goals, profile, [Burgos *et al.*, 2007].

The games and simulations are conversational adventures which are both usable and understandable. These characteristics benefits to the content creator, the professor in most cases, as long as to the final user, usually the student [Moreno-Ger *et al.*, 2008; Torrente *et al.*, 2008].

The software system is composed of several interconnected modules which allow the integration and intercommunication between games and simulations and several tools widely used for communities of professors. Initially these tools are the learning management systems Moodle and .LRN, and the authoring and learning units execution system LAMS [Burgos *et al.*, 2006; Moreno-Ger *et al.*, 2006]. In this context the Meta-Mender Recommendation System will suggest to users the best learning path according with their preferences, goals and objectives and will help with the game adaptation problem. As an example, the following meta-rule allows the generation of rules for the forums that the user has access to, see Figure 4.

```
rule ForumRecommendationRule
salience -1
when
  TelmaUser($userId : userId, $userName :
    userName, $mainInterest:
    mainInterest)
  TelmaForum($forumId : forumId,
    $mainTopic : mainTopic)
  TelmaForumAccessPerStudent(forumId ==
    $forumId, userId == $userId)
then
  WriteForumRecommendationRules(filePath,
    $userId, $userName, $forumId,
    $mainTopic);
end
```

Figure 4. A Meta-rule from Telma Application

This kind of rules allows for adaptation in case of the addition of a new forum to the application, a fact that can happen at any moment.

```
modify(amount->"0.5";
    comment->"Adjusting originality
    rating (by 0.5) for high ratings
    of other albums by this artist.";
    variable->originality;
    product->?item)
  :-
    rating(itemID->?item2;
    originality->"9.0"!?REST0),
    product(itemID->?item2;
    artist->?artist!?REST1),
    product(itemID->?item;
    artist->?artist!?REST2).
```

Figure 5. A Modify Rule from the RACOFI System

5 Expressive Power of Meta-rules

In this section we provide an example of the expressiveness power of meta-rules. We expose how a large set of rules can be generated from some few meta-rules.

```
tax(amount->"%15";
    comment->"15 percent HST")
  :-
    location(nb).
```

Figure 6. A Tax Rule from the RACOFI System

To this extend, we lean on the set of rules published on [Anderson *et al.*, 2003]. This rule-set contains 20 rules that modify user ratings based on item similarity. An example of these modify rules can be seen on Figure 5.

Other set of rules are the tax rules, there are 20 tax rules in the referenced paper. See Figure 6 for an example of these rules.

Finally the last set of rules is the so-called: `NotOffered` rules. A sample of these rules can be seen on Figure 7. There are 12 such rules in the rule-set.

```
NotOffered(itemID->?itemID)
  :-
    userLevel(beginner),
    product(itemID->?itemID;
      impression->?IMP!?REST),
    $lt(?IMP,7,true).
```

Figure 7. A NotOffered Rule from the RACOFI System

So the RACOFI rule-set defines more than 50 rules. This number is not very high but its maintainability can consume a lot of time. Also it is very hard to confirm that the rule-set is consistent with current data, suppose that it is necessary to modify a tax or a rating, it would be necessary to traverse the affected rules to check that everything is correct. Also on the fly rule generation, in order to increase adaptation, is a feature not easily covered with a static rule-set.

```
rule modifyMetarule
  when
    RatingAmountPair($amount : amount,
      $rating : rating)
    ProductMetadata($metadata : metadata)
  then
    WriteModifyRule(filePath, $metadata,
      $amount, $rating);
end

rule taxMetarule
  salience -1
  when
    TaxData($amount : amount,
      $location : location)
  then
    WriteTaxRule(filePath,
      $amount, $location);
end

rule notOfferedMetarule
  salience -2
  when
    NotOfferedData($maximum : maximum,
      $metadata : metadata, $student
      student, $userLevel : userLevel)
  then
    WriteNotOfferedRule(filePath,
      $maximum, $metadata, $student,
      $userLevel);
end
```

Figure 8. A Meta-rule-set that Generates the RACOFI Rule-set

The meta-rule approach discussed in this paper and implemented in our Meta-Mender Recommendation System is very useful to solve the problems mentioned above. Identifying common rule´s structure it is possible to write a concise meta-rule-set able to generate on the fly any

number of rules. This metadata, because meta-rules usually gets metadata as input, driven rule generation help solving the consistency problem, because rules can be regenerated at any moment after a change in the metadata.

Adaptation is also enhanced because different rules can be generated depending on user goals, needs and interests. As a side effect the development time of a rule-set is drastically reduced as long as a large number of rules can be generated with only one meta-rule.

As an example a set of meta-rules able to generate the whole rule-set of the RACOFI system is shown in Figure 8.

```
function void WriteModifyRule(
  String filePath, String metadata,
  String amount, String rating)
{
   FileWriter output = new
     FileWriter(filePath, true);
   MessageFormat mFormat = new
     MessageFormat("");
   output.write(mFormat.format(
     "modify(amount-> \"{0}\";\n",
     new Object[]{amount}));
   output.write(
     mFormat.format(
     "        variable->{0};\n",
     new Object[]{metadata}));
   output.write("        product->?item)\n");
   output.write("    :-\n");
   output.write(mFormat.format(
     "       rating(itemID->?item2;" +
     "{0}-     >\"{1}\"!?REST0),\n",
     new Object[]{metadata, rating}));
   output.write("       product(" +
     "itemID->?item2;artist-" +
     ">?artist!?REST1),\n");
   output.write("       product(" +
     "itemID->?item;artist-" +
     ">?artist!?REST2).\n");
  output.close();
}
```

Figure 9. The WriteModifyRule Function

In Figure 9 it can be seen a function that generates the modify rule-set of the RACOFI System. The rest of the functions referenced in Figure 8 are similar to this one.

6 Conclusions and Future Work

In this paper, we present a meta-rule based approach for rule generation. Meta-rules are rules that generate rules, and are able to generalize a rule-set providing bases for adaptation, reengineering and on the fly generation. In this paper, the benefits of meta-rules have been exposed. As an implementation example some details of Meta-Mender a meta-rule based Recommendation System have been also presented. The Meta-Mender Recommendation System is a component of, at present, two educational applications in development and test. The concept the Meta-Mender is based on, meta-rules for adaptation starts a new branch in the recommendation field and broadens the scope for new solutions in the field.

Meta-rules are an effective solution to provide a personalized recommendation to the learner, and constitute a new approach in rule-based Recommendation Systems.

Meta-rules constitute a new abstraction level, which provides foundations for effective adaptive and personalized processes, such as in, e.g. learning. This abstraction level is also highly valuable for adaptation. Rules can be different for different users or for the same user at different periods of time. Meta-Mender is a recommendation system that implements this approach.

At present, we use Meta-Mender in two R&D projects. In both, the next step is an evaluation phase with real data from actual users (in fact, two different, separate target groups). These two evaluation processes will provide a first-hand feedback of the implementation of Meta-Mender. Out of these results, we will refine the engine and we will design a visual authoring tool for meta-rules.

Acknowledgments

The research presented in this paper has been partially supported by the following projects of the Plan Avanza, a Spanish, nationally funded R&D programme: FLEXO (www.ines.org.es/flexo, TSI-020301-2009-9), GAMETEL (www.ines.org.es/gametel, TSI-020110-2009-170), TELMA (www.ines.org.es/telma, TSI-020110-2009-85).

References

[Abel *et al.*, 2008] F. Abel, I.I. Bittencourt, N. Henze, D. Krause and J. Vassileva. A Rule-Based Recommender System for Online Discussion Forums. *Proceedings of the 5th International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, Hannover, Germany, 2008.

[Anderson *et al.*, 2003] Michelle Anderson, Marcel Ball, Harold Boley, Stephen Greene, Nancy Howse, Daniel Lemire and Sean McGrath. RACOFI: A Rule-Applying Collaborative Filtering System. *Proceedings of the IEEE/WIC COLA'03*, Halifax, Canada, October, 2003.

[Burgos *et al.*, 2006] D. Burgos, C. Tattersall and R. Koper. Re-purposing existing generic games and simulations for e-learning. *Special issue on Education and pedagogy with Learning objects and Learning designs*. *Computers in Human Behavior*, 2006.

[Burgos *et al.*, 2007] D. Burgos, C. Tattersall and R. Koper. How to represent adaptation in eLearning with IMS Learning Design. *Interactive Learning Environments*, 15(2), 161-170, 2007.

[DROOLS] DROOLS. The Business Logic Integration Platform, http://www.jboss.org/drools.

[El Helou *et al.*, 2009] S. El Helou, C. Salzmann, S. Sire and D. Gillet. The 3A contextual ranking system: simultaneously recommending actors, assets, and group activities. *In Proceedings of the Third ACM Conference on Recommender Systems RecSys '09 ACM*, New York, New York, USA, 373-376, 2009.

[Forgy, 1982] C. Forgy. Rete: A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem, *Artificial Intelligence*, 19, 17–37, 1982.

[Herlocker, 1999] J. L. Herlocker. Position Statement - Explanations in Recommender Systems. *In Proceedings of the CHI' 99 Workshop*, Pittsburgh, USA, 1999.

[McCarthy *et al.*, 1985] J. McCarthy, P. W. Abrahams, D. J. Edwards, T. P. Hart, and M. I. Levin. LISP 1.5 Programmer's Manual, MIT Press, 1985.

[Moreno-Ger *et al.*, 2006] P. Moreno-Ger, I. Martínez-Ortiz, J. Sierra and B. Fernández-Manjón. A Descriptive Markup Approach to Facilitate the Production of e Learning Contents. *In Proceedings of 6th International Conference on Advanced Learning Technologies* (ICALT 2006), 19-21, Kerkrade, The Netherlands, (IEEE Computer Society), 2006.

[Moreno-Ger *et al*., 2008] P. Moreno-Ger, D. Burgos, I. Martínez-Ortiz, J. Sierra and B. Fernández-Manjón. Educational Game Design for Online Education. *Computers in Human Behavior* 24(6), 2530–2540, 2008.

[Rocchio, 1971] J.J. Rocchio. Relevance feedback in information retrieval, in the SMART Retrieval System. *Experiments in Automatic Document Processing*, Englewood Cliffs, NJ. Prentice Hall, Inc., 313-323, 1971.

[Torrente *et al*., 2008] J. Torrente, P. Moreno-Ger, and B. Fernández-Manjón. Learning Models for the Integration of Adaptive Educational Games in Virtual Learning Environments. *In Proceedings of the 3rd International Conference on E-learning and Games*, Nanjing, China. Lecture Notes in Computer Science 5093, 463–474, 2008.

Social IPTV: a Survey on User-Acceptance

Daniel Schreiber
Telecooperation Lab

Osama Abboud, Sandra Kovacevic
Multimedia Communications Lab

Andreas Hoefer, Thorsten Strufe
Peer-to-Peer Networks

TU Darmstadt
lastname@{tk.informatik,kom,cs}.tu-darmstadt.de

Abstract

Incorporating Social Networking and IPTV, the two arguably fastest growing and most accepted services on the Internet today, yields strong synergies. However, it opens a very complex design space of feature combinations. Selecting features, aiming at achieving the best user acceptance, hence, is a difficult, yet vital task for the development of an integrated service. This paper presents the results of an initial user study conducted to gain a better understanding of the complex design space. It identifies classes of demanded, promising features and indicates that *social features* in IPTV services in general will be well accepted, even if they are quite immersive to the TV experience. The study was conducted with user groups from central Europe as well as with a group from Korea.

1 Introduction

IPTV, i.e., broadcasting of multimedia streams using well understood internet protocols, is a fast spreading technology for distributing multimedia TV content to consumers. With the rapid growth of the number of broadband links to the home as well as the available bandwidth at mobile devices, the connectivity of the consumer is constantly increasing, which opens the opportunity for innovative new services combining multimedia streams with interactive features.

Besides IPTV, online social networking (OSN) services like Facebook etc. have been some of the most popular, fastest growing internet services in the last years. Thus, we conjecture that one promising approach to develop an innovative online service is *Social IPTV*, the integration of OSN and IPTV functionality into one consolidated service.

Integrating both, poses to be a difficult task, since their utilization properties are located at two different extremes of participation: Currently, the TV experience is characterized as being passive. However, in order to benefit from social TV, users will have to interact with the content as well as to communicate and collaborate with other viewers. Hence, the selection of integrated services and applications is non-obvious, since it needs to gauge between the comfort of passive content consumption and active contribution. In this paper we report initial results of an initial survey we conducted, which corroborate our hypothesis: Users will indeed accept and appreciate Social IPTV services. Furthermore, the user study revealed which features of OSN and IPTV integration are most interesting to the user.

2 Acceptance of Social IPTV Features

Although it may seem natural to integrate OSN features into IPTV – e.g., for automatic recommendations and adapting the TV program to the user at hand [Vildjiounaite *et al.*, 2008]; for creating a social viewing experience [Nathan *et al.*, 2008]; or for joint commenting [Cattelan *et al.*, 2008] – we wanted to make sure that this actually corresponded with user needs. Also, we wanted to find out which OSN features would be most valuable to end-users. To clarify these questions, we performed a two staged user study.

2.1 Step 1 - Focus Group Interview

First, we conducted an in-depth interview with a focus group to gather ideas for potential Social IPTV services.

Method The interview has been directed by one interviewer and minutes were both taken together at a white board and additionally by one observer of the interview. The focus group consisted of 20 technology savvy people with academic background. The interview has been conducted in three phases and took just under an hour. The first phase consisted of free brain storming, in which the group was encouraged to suggest any possible features, applications or scenarios in the context of Social IPTV. All ideas and innovations were recorded on the white board in order of suggestion. The suggestions were ordered and grouped in the second phase of consolidation, in which the participants were encouraged to extend or further detail some of the ideas as they were discussed. Finally, the participants were asked to roughly estimate innovation, degree of attractiveness, and technical feasibility of each item in the list in the last phase.

Results The results of this interview were about 30 distinct proposed applications. Several of these did not include social aspects, but only relied on the additional flexibility of general IPTV, e.g., feedback, online database integration, and targeted advertising. The applications that include social aspects can be grouped into five clusters.

Content Recommendations Using information about what friends are currently watching or used to watch to provide viewing suggestions.

Community Awareness Displaying information about who is watching the same program, who is watching TV at all to create a more social TV experience.

Community Meta Content Comments and annotations of IPTV content, e.g., alternative opinios for news, annotating glitches in movies.

End-2-End Communication Live communication with friends viewing the same content, e.g., text and video chat.

Participatory IPTV Remixing of media streams, e.g., providing an alternative audio comment to a selected group of friends

Social Applications Providing interactive applications to a community, e.g., polls, betting, or visual annotation.

Discussion The results are grouped by *immersiveness*, i.e., adapting TV program recommendations based on the user's profile in the OSN leaves the user completely passive – input is provided only implicitly by letting the system analyze the viewing behavior of the user and its friends in the OSN, thereby automatically creating a user model. Content recommendation based on viewer behavior and user models has, e.g., been suggested by [Vildjiounaite *et al.*, 2008].

Community awareness can be generated by ambient displays, e.g., as proposed in [Harboe *et al.*, 2008] or buddy list [Boertjes, 2007]. Thereby, it is important that the realisation is adapted to the user at hand, her personal preferences as well as the current viewing context.

Community generated meta content is extremely popular in OSN, where users can, e.g., comment on other users, or take part and comment on virtual events. Such features for IPTV have been proposed in [Nathan *et al.*, 2008]. Here, the content from the community is interwoven with the IPTV content and users have to explicitly contribute this content. Even more demanding in terms of user attention is live communication, e.g., by chat [Abreu *et al.*, 2001], voice chat [Coppens *et al.*, 2004], or video chat [Abreu *et al.*, 2001].

The next two clusters represent even more immersive acts for the user. Instead of just using features provided by the system, the user becomes a content developer, i.e., providing his own IPTV content through live-streaming [Stickam Social Video Streaming,], or content editing [Cattelan *et al.*, 2008], or even providing applications, leveraging end-user development.

This analysis shows that most of the applications have already been envisioned by existing projects. However, we are not aware of a systematic assessment whether which degree of immersiveness would actually be tolerated by the users, e.g., whether the added benefit of the applications would be worth the disturbance.

2.2 Step 2 - Questionnaire Survey

As a next step, the acceptance of several Social IPTV applications in an European and Korean audience has been tested in a survey. Three application scenarios from the focus group interview and a general scenario of offering a "Social Electronic Program Guide", representing the Content Recommendation cluster, have been selected for the survey.

Participants were asked to state their feeling towards the importance of the problem presented in the scenario as well as to rate the envisioned solution. The four selected scenarios were the "SportsPub" (allowing a virtual crowd to watch a sports event together, connecting spectators via video and text chat), shared highlighting (collaborative, shared highlighting of areas on the screen, e.g., to bet where the next goal is scored), the "PlayerStats" (the possibility to add annotations to certain objects on screen), and the "Social Electronic Program Guide". All scenarios follow the line of thought of offering a collaborative

After finishing watching the news, Sue can't decide what program to watch next. She activates the Social Electronic Program Guide to see what her friends are watching. Her friend Mick is following the news on the election day in "Marx' Corner", a group that Engin, another friend, has set up. Other friends are watching sports together and some have tuned into Casablanca...

Figure 1: Stimulus for the "Social Electronic Program Guide" scenario.

TV experience, including the possibility to interact in tele presence as well as allowing for both textual and graphical annotation. The notion of allowing users to participate by creating their own applications has been woven into the "PlayerStats" scenario.

Participants The questionnaire was administered to three different groups of participants: Group A - audience of the World Usability Day ($n = 32$, 4 female, 1 unstated); group B - students in a university course ($n = 44$, 26, 0); group C - ETRI[1] staff ($n = 17$, 2, 1). User groups A and C are technology affine, while group B is more reluctant about new technology. Users in groups A and B are European, users in group C are from Korea.

The age of the users ranged from 15 over 60 years (group A: 1 person <20y, 18 persons: 20 - 29y, 10: 30 - 39y, 1: 40 - 49y, 1: 50 - 59y, 1 unstated, group B 2 <20y, 40: 20 - 29y, 1: 40 - 49y, 1: >59, group C: 1 <20y, 6: 20 - 29y, 9: 30 - 39y, 2: 40 - 49y). The amount of TV consumption differed between the three groups. In group A, only 40% stated they were watching TV more than 4h per week. For the other groups the values were 82% (B) and 61% (C). We believe that the reason for this is that the users in group A spent significantly longer time surfing the web compared to the other groups, we did not test this though. 47% - 52% in each group stated that they talked about what they watched on TV with their friends.

Method In the questionnaire, each scenario was presented as a combination of an illustrating picture together with a short, illustrative description of an exemplary situation, in which the application would be used (cmp. fig. 1, and 2). The participants were confronted with the same set of questions for each scenario.

[1]Electronics and Telecommunications Research Institute, a Korean research institute

The questions were grouped in three parts: two questions to generally test if the participants had completely read and understood the scenario, two questions to test the participants' valuation and degree of esteem for the scenario, and an open question for comments or further ideas. The closed questions were designed offering four possible answers as a four-level Guttman Scale in order to avoid neutral answers. Following the four scenarios the participants had the opportunity to provide further comments or ideas on the topic. The questionnaire ends with a set of statistic questions for the sample description.

Even viewers can create new applications and share them with their friends: Lyn created a small application to tag objects in the show with some comments. She is a big fan of her team and especially likes Pepitto, the youngest player ever to play in the world cup! She knows all about him and shares her knowledge with the other viewers of Eiko's SportsPub.

Figure 2: Stimulus for the "PlayerStats" scenario.

Results All four scenarios and applications were generally perceived positively by all participants, in general all promise to make watching TV more fun. The three user groups had some characteristic differences, which can be seen in figure 3. Group B was less enthusiastic about the scenarios and solutions but thought that watching TV would be more fun and more interesting with the proposed enhancements. Group C was very enthusiastic but was more conservative in the rating of whether watching TV would be more fun and more interesting. Group A takes a middle position compared to groups B and C. A major concern consistenly mentioned in the responses to the open questions was that the system needs to preserve the user's privacy and there should be the possiblity to *mute* social features in order to be able to watch certain shows without distraction.

Summarising, the idea of providing an interactive, Social IPTV service has been accepted by our study sample, however users are not overly enthusiastic. Likewise, the three application scenarios have found some resonance in the study groups, and we think it makes sense to further investigate what users liked about them and how they can be improved.

3 Discussion

Oksman et al. have performed a study, which corresponds to a single aspect of our work: users in [Oksman and others, 2009] where asked, if they were interested in chatting with each other while watching TV. Our fundamental results are supported by their conclusion that users consider watching TV as an inherently social activity. The study additionally backs the requirement of our users that privacy and the possibility to disable interactive features is vital for the acceptance of a social IPTV service.

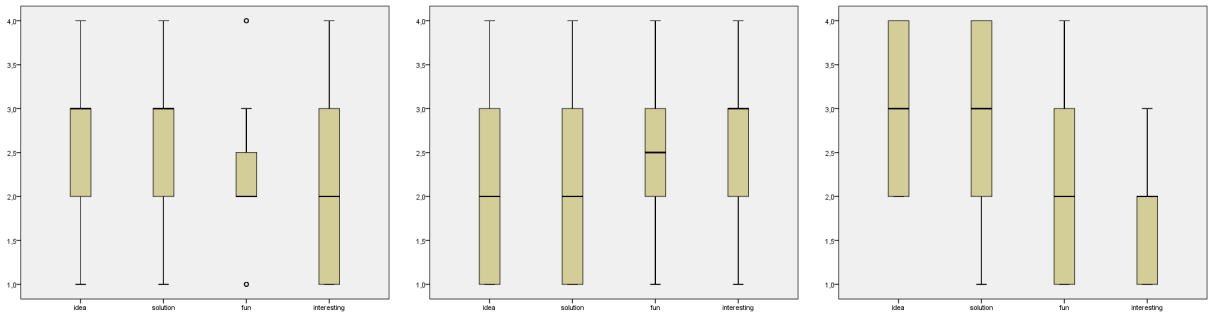
Given the evidence in existing studies on social TV and related topics (e.g., [Oksman and others, 2009] and [Harboe *et al.*, 2008]), as well as our own results it seems clear that social TV applications will be accepted by the user. This poses the question about how such applications should be built, i.e., which infrastructure is necessary to support such applications at runtime.

The most important question in our opinion is how to generate the critical mass that is necessary to support social TV features. The value of social applications to the user increases with the total number of users taking part in the OSN. Ramping up on the millions of users of OSN like Facebook or StudiVZ would make OSN applications highly valuable right from the start. To foster such kind of applications, extensions to existing well established IPTV platforms are needed. Existing solutions, like, e.g., the MHP [Piesing, 2006] API, currently do not forsee dedicated support for integrating OSN.

The drawback of current OSN is that all information (profiles, friend relations) is stored on the servers of a provider. This introduces a severe security risk, which has received much attention recently [Bilge *et al.*, 2009]. Privacy concerns are relatively important to IPTV users. However, IPTV systems incorporating set-top boxes installed at the user's home open up the opportunity to create a new kind of online social network based on peer-to-peer principles that does not come with these security limitations [Cutillo *et al.*, 2009]. These set-top boxes' online times are longer and more predictable (i.e., until the end of the current show) than those of nodes in traditional peer-to-peer networks. Therefore, one might have the privacy advantage of a peer to peer based OSN, without the drawbacks of low availability of current research peer-to-peer OSNs.

Therefore, another option would be to use IPTV technology to create a new Thus, the valuable data from OSN can be used to enhance the TV experience without putting the user's privacy at risk. An additional benefit of this is that people who are not technology savvy are much more likely to join these social networks, which provides a much richer data basis for recommender systems.

Inevitably, social TV will be interactive, i.e., users need means to provide community content, to chat with each other and even to do online remixing of streams. Thereby, the interaction with the TV should easily be scalable from a completely passive mode, where the user just consumes content; over a semi-interactive mode, where the user can, e.g., vote with the help of her TV remote; up to a higly interactive mode, where sophisticated input devices are used. It must be easy to switch between these different modes easily, requiring a much more flexible setup for in- and output devices than with current IPTV platforms. In this regard, we see a need for adapatation of the interface to the user and her needs. One approach is the use of stereotypes that allow the user to change between different levels of immersiveness, e.g., ranging from a completely passive



(a) Group A (b) Group B (c) Group C

Figure 3: Results from the questionnaire for the three user groups. The median is marked by the bold line, the interquartile range with the box, max and min values by the whiskers. The first data point in each chart refers to how interesting the idea of the scenario was rated; the second data plot refers to how well the solution was appreciated; data point three and four relate to the answers whether the solution would make watching TV more fun and more interesting.

traditional TV experience, to a completey interactive experience, which is currently knwon from the web used on normal PCs.

4 Summary and Outlook

This paper presented the results of a user study on the acceptance of Social IPTV, which is a combination of social networks and IPTV features. It introduces some social IPTV scenarios, which have been generated from interviews with focus groups. A subsequently performed surbey supports the hypothesis that social IPTV applications will be accepted by a central European as well as a Korean audience. Even immersive features, like content creation and end-user development, have been received very well and the results show that they will potentially be used. This is a strong argument for developing and providing the necessary infrastructure to implement such applications, since current platforms lack dedicated support for implementing Social IPTV applications. Although the study confirms that there is a potential benefit from integrating OSN features with IPTV, it did not reveal too much detail what the most interesting features would be. The focus interview resulted into an initial feature set. However, we are currently conducting further studies with working prototypes to find out which features or classes of features are accpeted by the end-user, and even more important, why these features are accepted.

The outstanding difference between traditional TV and Social IPTV is the move from a passive TV experience, to a more immersive experience. Users expressed the wish to be able to return to the completely passive mode. We think this can be generalized, and the final Social IPTV system should allow for transition from completely passive to fully immersive interfaces. To implement this, we want to identify certain user stereotypes to which we can adapt the interface.

Achknowledgements

We thank the Korean ETRI for supporting our research.

References

[Abreu *et al.*, 2001] Jorge Abreu, Pedro Almeida, and Vasco Branco. 2beon - interactive television supporting interpersonal communication. In *Multimedia 2001, Proceedings of the Eurographics Workshop*. Springer, 2001.

[Bilge *et al.*, 2009] Leyla Bilge, Thorsten Strufe, Davide Balzarotti, and Engin Kirda. All Your Contacts Are Belong to Us: Automated Identity Theft Attacks on Social Networks. In *18th Intl. World Wide Web Conference (WWW'09)*, pages 551 – 560, 2009.

[Boertjes, 2007] Erik Boertjes. Connectv: Share the experience. In *EuroITV - Workshop: Social Interactive Television*, 2007.

[Cattelan *et al.*, 2008] Renan G. Cattelan, Cesar Teixeira, Rudinei Goularte, and Maria Da Graça C. Pimentel. Watch-and-comment as a paradigm toward ubiquitous interactive video editing. *ACM Trans. Multimedia Comput. Commun. Appl.*, 4(4):1–24, 2008.

[Coppens *et al.*, 2004] Toon Coppens, Lieven Trappeniers, and Marc Godon. Amigotv : towards a social tv experience. In *EuroITV*, 2004.

[Cutillo *et al.*, 2009] Leucio-Antonio Cutillo, Refik Molva, and Thorsten Strufe. Safebook: a Privacy Preserving Online Social Network Leveraging on Real-Life Trust. *IEEE Communications Magazine*, December 2009.

[Harboe *et al.*, 2008] Gunnar Harboe, Crysta J. Metcalf, Frank Bentley, Joe Tullio, Noel Massey, and Guy Romano. Ambient social tv: drawing people into a shared experience. In *Proceeding of CHI*, pages 1–10, New York, NY, USA, 2008. ACM.

[Nathan *et al.*, 2008] Mukesh Nathan, Chris Harrison, Svetlana Yarosh, Loren Terveen, Larry Stead, and Brian Amento. Collaboratv: making television viewing social again. In *UXTV*, pages 85–94, New York, NY, USA, 2008. ACM.

[Oksman and others, 2009] Virpi Oksman et al. Tv is just one of the screens at home. consumers and changing tv watching. In *EuroITV*, 2009.

[Piesing, 2006] J. Piesing. The dvb multimedia home platform (mhp) and related specifications. In *Proceedings of the IEEE*, volume 94, pages 237–247, 2006.

[Stickam Social Video Streaming,] Stickam Social Video Streaming. Online: http://www.stickam.com/.

[Vildjiounaite *et al.*, 2008] Elena Vildjiounaite, Vesa Kyllönen, Tero Hannula, and Petteri Alahuhta. Unobtrusive dynamic modelling of tv program preferences in a household. In *EuroITV*, pages 82–91, 2008.

Using a Semantic Multidimensional Approach to Create a Contextual Recommender System

Abdulbaki Uzun
Service-centric Networking
Deutsche Telekom Laboratories, TU Berlin
abdulbaki.uzun@telekom.de

Christian Räck
Competence Center FAME
Fraunhofer Institute FOKUS
christian.raeck@fokus.fraunhofer.de

Abstract

Item recommendations calculated by recommender systems mostly in use today, only rely on item content description, user feedback and profile information. In modern mobile services, however, contextual information and semantic knowledge can play a significant role concerning the quality of these recommendations. Therefore, the *SMART Recommendations Engine* of Fraunhofer FOKUS is extended by the *SMART Multidimensionality Extension* and the *SMART Ontology Extension* that enable the recommender to incorporate contextual and semantic data into the recommendation process. The demonstration of the *SMART Ontology Extension* visualizes that the preciseness of recommendations can be increased by exploiting implicit and indirect knowledge, classification and location information gained from ontologies when generating recommendations in the scope of an exemplary food purchase scenario.

1 Introduction

Today, people are confronted with a large amount of information in the World Wide Web [Shenk, 1998]. Receiving a small subset of desired and filtered content through standard search engines turns out to be very difficult. And it becomes quite impossible, if user specific needs and interests should be taken into consideration.

Recommender systems handle this issue by filtering relevant information and providing personalized content recommendations to users based on their profile and feedback. Numerous recommendation methods were designed over the years to improve the accuracy of recommendations. The most popular ones are content-based and collaborative filtering algorithms or hybrid approaches comprising both them [Adomavicius and Tuzhilin, 2005].

The results delivered by hybrid methods are often acceptable. However, they only depend on content descriptions and ratings given to items, and user profile information. In a time when mobile and location-based services become very popular, context information and semantic knowledge can play a decisive role in order to significantly improve the preciseness of personalized recommendations. If John, for example, is vegetarian, eats only organic food, tries to live economical and goes shopping nearby, it does not make sense to recommend him groceries in stores far away or only in discounters without taking his preference for vegetarian and organic food into consideration. This example shows that context as well as semantic information (e.g. implicit knowledge about which food products fit to certain eating preferences) are important to satisfyingly answer a user's grocery recommendation request.

In order to harness the potential of contextual and semantic information, the generic recommender system of Fraunhofer FOKUS, the *SMART Recommendations Engine* [Raeck and Steinert, 2010], has been extended by two new recommender extensions. Inspired by the work of [Adomavicius *et al.*, 2005], the *SMART Multidimensionality Extension* enhances the two-dimensional matrix representation of recommender data by a multidimensional recommendation model allowing the integration of additional contextual information when generating recommendations. The *SMART Ontology Extension*, on the other hand, enables the recommender to incorporate contextual and semantic information gained from ontologies (e.g. implicit and semantic knowledge about a user and his preferences, location, time or ontological classification information). For this purpose, the extension provides a tool to exploit semantic data stored in ontologies and perform context and semantic filtering on the recommender database using several filters. Both extensions can be used independently from each other or together depending on the given scenario and application.

This paper is organized as follows: First, an overview about related work in the field of context-aware and semantic recommender systems is presented. Afterwards, a background about the *SMART Recommendations Engine* is given. Following that, concepts for the *SMART Multidimensionality Extension* are described. Section 5 explains the *SMART Ontology Extension* including the automated mapping of ontological information into the recommender's data model. In section 6, the functionality of the *SMART Ontology Extension* is demonstrated within the scope of a food purchase scenario.

2 Related Work

Contextual and semantic information is incorporated in the field of recommendations in various ways. A context-aware collaborative filtering system is presented by [Chen, 2005], which generates item recommendations for a user based on different context situations. In order to achieve that, the traditional collaborative filtering is extended, so that feedback of like-minded users in a similar context can be used to recommend items to the active user in his current context.

[Adomavicius *et al.*, 2005] propose a multidimensional recommendation model, in which additional contextual di-

mensions can be added to the traditional *User x Item* matrix representation. Recommendations are calculated by using the *reduction-based approach*, which reduces the problem of multidimensional recommendations to a traditional two-dimensional matrix in the required context, so that widely known traditional recommendation techniques can be applied after the reduction is done.

Another methodology is suggested by [Farsani and Nematbakhsh, 2006], which recommends semantic products to customers in the context of eCommerce based on product and customer classification via OWL.

[Kim and Kwon, 2007], on the other hand, developed an ontology model with a multiple-level concept hierarchy for a grocery store scenario with four different ontologies: a *product*, *location*, *consumer* and *record* ontology. From the *product* ontology, most relevant products are taken using user information modeled in the *consumer* and *record* ontology. Recommended products are presented in a concept hierarchy ranging from most specific to most broad. When users select some of these concepts, the context of the request is subsequently refined enhancing the specificity of the provided results.

Another interesting approach is demonstrated by [Yu *et al.*, 2006]. They present a context-aware media recommendation platform called *CoMeR*, which uses an OWL ontology context description, a *N x M*-dimensional model and a hybrid processing approach to support media recommendations for smart phones.

In another paper by [Setten *et al.*, 2004], the integration of a context-aware recommender system into the mobile tourist application *COMPASS* is described, where users get touristic information and services recommended based on their interests and contexts.

COReS, which stands for *context-aware, ontology-based recommender system for service recommendation*, is another example. It was developed by [Costa *et al.*, 2007] and this recommender system extends the capabilities of the *INFRAWARE* [Pereira Filho *et al.*, 2006] service platform by supporting service selection and user needs satisfaction for a certain context.

Previous research activities are either focused on the integration of context or semantic information. However, incorporating both – context information and semantic domain knowledge – would increase the preciseness of recommendations decisively. The food scenario shows that the integration of both types of data is necessary to satisfyingly answer a recommendation request. For example, in order to generate accurate grocery recommendations, the system has to be aware of the location of the user (context) and has to be capable of using implicit knowledge in order to relate the user's eating preferences to the right groceries (semantic information). That's why, Fraunhofer's engine was extended using both types of data.

3 SMART Recommendations Engine

The *SMART Recommendations Engine* is a generic recommender system that enables internet businesses, rich media and entertainment services or SMEs deliver a more personalized experience by providing recommendations in their respective application domain. By offering a flexible, general purpose algorithmic model, the engine makes it possible to formulate application specific recommendation algorithms. These algorithms and the application specific data model are declared at configuration time by assembling selected components. By adding custom components through the provided API, the capabilities of the recommender can be extended in order to meet specific application demands. These custom components can be built from functional groups, such as basic mathematical operations, similarity and relevance computations, sorting and filtering, and data access.

In the system, data is represented in an entity-relationship-like data model. An entity type that includes a set of entities is named *domain*, whereas relations between domain entities are represented by *matrices* (see Figure 1). A user domain, for example, can contain all users, while an item domain can comprise all items in a specific application. Ratings given by a user to a certain item can be represented by a rating matrix, whose rows and columns are associated to the these domains.

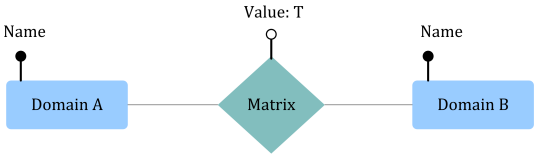


Figure 1: Recommender Data Model

Recommendation algorithms calculate relevance values for each *User x Item* pair. Differing on the given application, the recommendation algorithm is assembled at runtime configuration by defining a network of matrix transformation components (e.g. a *similarity* computation component). The matrix operations are applied on a data set in a hierarchical manner leading to the estimated utility function at the top node of the tree. Figure 2 shows such a generic algorithm hierarchy, where the nodes represent matrix operations and data is propagated along the edges in form of matrices.

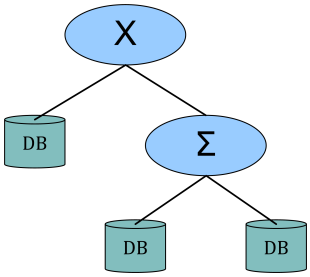


Figure 2: Generic Algorithm Hierarchy

The engine also provides a custom query language called *Sugar Query Language (SuQL)*, which is used to request recommendations and related data at runtime. Via *SuQL* various recommendation algorithms can be selected and combined. Constraints on the item set to be recommended can also be specified, so that only items that fulfill these constraints are included in the final recommendation. For this purpose, the recommender already offers a variety of sorting and selection filters, which can be used to alter the result set by certain properties.

One example for a filter is the *Proximity Filter*, which in combination with a Geo-Location typed domain, an *Item x Location* matrix, a given center position and a maximum range, is capable of selecting items (e.g. shops) in a given geographic region. By using the lookup capabilties of the *SMART Ontology Extension* (see section 5), items

can be filtered based on location constraints, like finding all grocery products sold in shops located within a given distance from the user's current location.

4 SMART Multidimensionality Extension

Contextual information can be exploited in the recommendation process by enhancing the level of dimensions from the traditional two-dimensional paradigm to multiple dimensions. Therefore the *SMART Recommendations Engine* is upgraded by the *SMART Multidimensionality Extension*, which is able to handle multidimensionality and hence provide more precise recommendations.

4.1 Enhanced Data Model

This extension enhances the generalized data model of the *SMART Recommendations Engine* for multidimensionality purposes by binding more than two domains to a rating matrix. Figure 3 shows the enhanced data model in an entity-relationship notation.

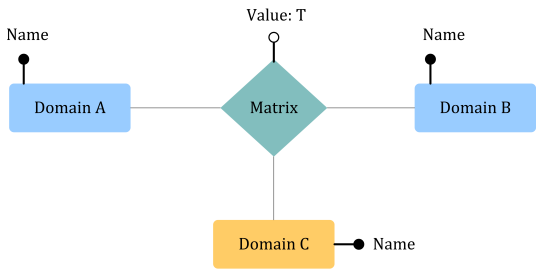


Figure 3: Enhanced Data Model

However, the recommender is designed to work two-dimensional; a matrix can only consist of two domains (one row domain and one column domain). That's why, the enhanced data model has to be adapted to the standard data model of the recommender. By merging several domains to one domain and hence mapping multiple dimensions to a two-dimensional matrix, the two-dimensional recommender data model can be kept with the advantage that multidimensionality features can be utilized at the same time (see Figure 4).

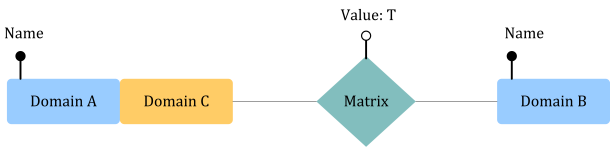


Figure 4: Merged Domains in the Enhanced Data Model

The two-dimensional matrix representation of multiple dimensions can be done through *serialization* by use of a hierarchical index. Each slice of the multidimensional cube is stored into one two-dimensional matrix making it possible to access the desired element with the help of an index (e.g. *Slice1.User#1*). Figure 5 shows a general MD-2D-Mapping for the enhanced data model.

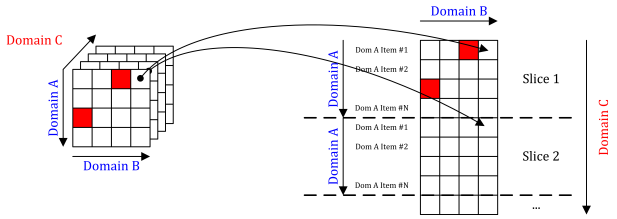


Figure 5: General MD-Mapping to 2D

4.2 Multidimensional Recommendation Algorithms

Various standard recommendation algorithms, such as user or item-based collaborative filtering, can be extended by the extension to multidimensional algorithms.

The standard *user-based collaborative filtering* algorithm applied by the *SMART Recommendations Engine*, for example, determines similar users by using ratings given for items by them. Based on this similarity and on the ratings given to items, predictions for new items are calculated. This algorithm can be upgraded to a *multidimensional user-based collaborative filtering* algorithm by merging several domains to one column domain.

Assume that there is a rating matrix *R[U :: C][I]* in the *User :: Context x Item* space, in which *User :: Context* displays users with a certain context.

$$R[U :: C][I] = User :: Context \times Item \tag{1}$$

The similarity between users in the same context is determined based on their ratings for items by applying a *similarity* transformation on *R[U :: C][I]*. This calculation also identifies similarities between users in different context situations, which can be useful for later recommendations.

$$Sim(R[U :: C][I], R[U :: C][I]) = User :: Context \times User :: Context \tag{2}$$

Having a similarity matrix *Sim[U :: C][U :: C]* and the rating matrix *R[U :: C][I]*, a *matrix product* transformation can be applied to calculate item predictions for users in a certain context.

$$PredictionP[U :: C][I] = Sim[U :: C][U :: C] * R[U :: C][I] = User :: Context \times Item \tag{3}$$

These predictions can be further processed and sorted to generate new recommendations for users in certain contexts.

To better illustrate the new *multidimensional user-based collaborative filtering algorithm*, an example query "*John wants to buy food for a soccer evening*" is used. This query includes three dimensions: the *user* (John), *food* and the *event* dimension (here: soccer evening). For that matter, the similarity between users, who bought food for a soccer evening, is calculated using the *User :: Event x Food* rating matrix twice for the similarity computation (see Figure 6).

By applying a matrix product transformation on the calculated similarity matrix *User :: Event x User :: Event* and the rating matrix, food predictions for John in the context of a soccer evening can be identified. These predictions can further be filtered and sorted based on John's eating habits, for example, and the rest result can

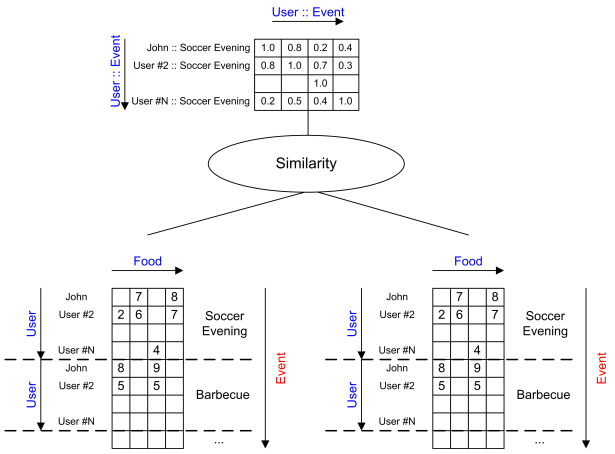


Figure 6: Exemplary Multidimensional User-based Collaborative Filtering Algorithm – Part 1

be presented as generated recommendations for John (see Figure 7).

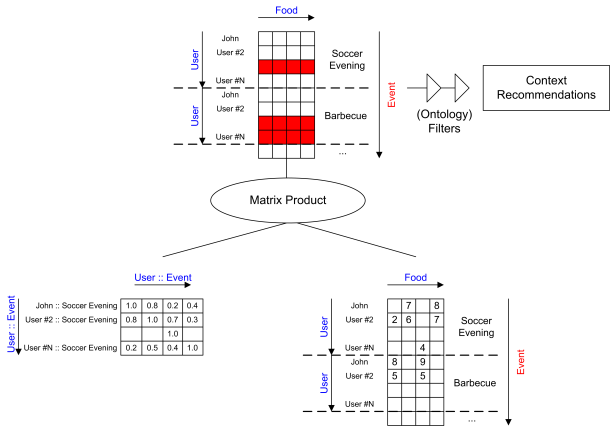


Figure 7: Exemplary Multidimensional User-based Collaborative Filtering Algorithm – Part 2

5 SMART Ontology Extension

Data that provides additional and useful information to the traditional *User x Item* representation, such as taxonomies, implicit and indirect knowledge about a user's preferences or location information can enhance the quality of recommendations.

Semantic web ontologies offer complex knowledge representation possibilites with which such semantic information can be modeled (the *Web Ontology Language (OWL)* [Dean and Schreiber, 2004]). Implicit knowledge, i.e. knowledge that is not directly included in an ontology, can be inferred using reasoning technologies.

The *SMART Ontology Extension* provides features with which the *SMART Recommendations Engine* is capable of exploiting semantic information from ontologies including information gained from reasoning mechanisms. In that way, implicit and indirect knowledge as well as taxonomies become available when generating recommendations.

The extension is divided into two main parts: The first part is the *Ontology Mapping*, where all relevant data available in pre-designed ontologies is mapped onto the data model of the recommender. The second part performs semantic filterings on the previously extracted data set using the *Ontology Filter* in order to generate semantic recommendations.

The functionality of the extension is explained using pre-designed ontologies for the food scenario. It comprises users with special eating habits (such as *vegetarian*, *vegan* or *organic*), who want to buy food from grocery stores nearby based on their eating preferences.

5.1 Ontology Mapping

The main constructs included in OWL ontologies are individuals, classes, a class hierarchy, object properties, datatype properties and restrictions. These constructs can be mapped onto the data model of the recommender, so that the recommendation engine becomes capable of handling ontology information.

Each class hierarchy in OWL is represented by a domain with the name of its respective root class (e.g. *Food*, *User* or *GroceryStore*). Individuals are items of a certain class and are mapped as items of their respective class hierarchy recommender domain. Furthermore, all classes in an ontology become elements of the general *Class* domain. While a *Food x Class* matrix, for example, shows to which classes a food product belongs to, taxonomies become clearly recognizable in a *Class x Class* matrix. Figure 8 shows an exemplary ontology class structure mapping for the food scenario.

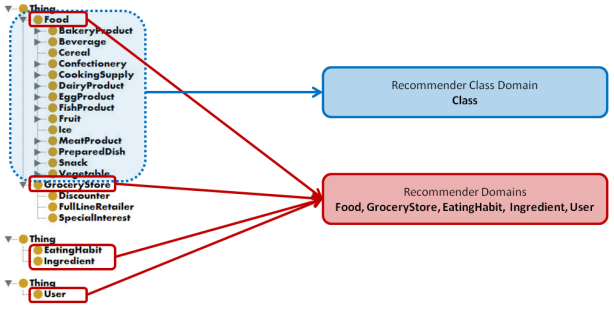
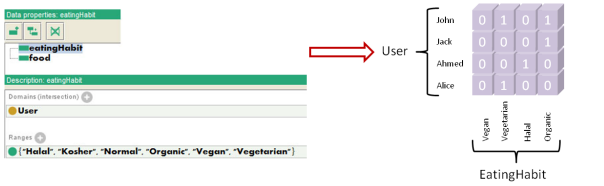


Figure 8: Root Class Representation

Object properties in OWL form a relation between individuals, whereas datatype properties build a relation between individuals and data types, such as *Integer*, *String* or *Boolean*. Matrices display relationships between domains and are therefore equal to object and datatype properties. The domain and range an OWL property has are both represented as domains in the recommender and their values are elements of these domains. If a property's domain or range is defined as an ontology class, the recommender domain name corresponds to the class's name as well. Otherwise, it corresponds to the property's name.

In the example of Figure 9, the property *eatingHabit* has a domain named *User* and data range values, such as *Vegan*, *Vegetarian* or *Halal*. Therefore the recommender domains are *User* and *EatingHabit* (name of the OWL property) defining a matrix. The values in the matrix are also directly mapped from the ontology and represent the assigned property values to individuals (e.g. *John eatingHabit Vegetarian*).

Based on the *Ontology Mapping* naming conventions, the *Ontology Mapping tool* automatically creates a recommender-compliant database including domains,



a) Datatype Property in the Ontology a) Matrix Representation of the Property

Figure 9: Datatype Property Representation

matrices and filters. Knowledge gained by reasoning mechanisms is also exported via the tool making it possible to exploit implicit knowledge in the recommendation process. As seen in the screenshot of the tool (Figure 10), the left part of the GUI represents the *Ontology Import* and *Ontology Export* functions, whereas the right part comprises all functions concerning the recommender.

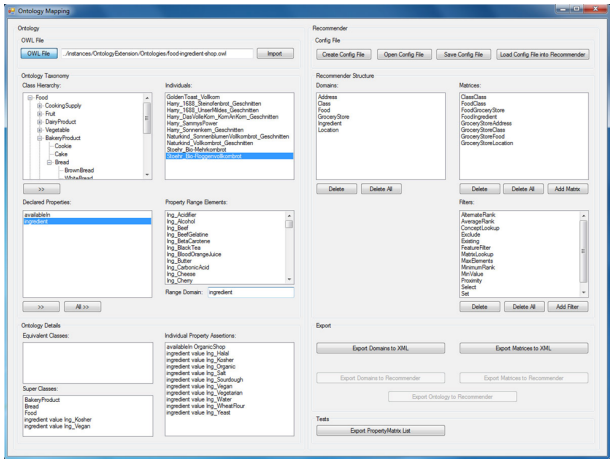


Figure 10: Ontology Mapping Tool

5.2 Ontology Filter

After successfully exporting ontology data into the recommender's database, the *Ontology Filter* can process the ontology information in the engine by performing a semantic filtering. This new filter extends the available filter list of the *SMART Recommendations Engine* and makes it possible to query implicit and indirect ontology knowledge at runtime.

Two different matrix operations are provided by the *Ontology Filter*, the *Concept Lookup* and the *Matrix Lookup*. In order to accomplish their task, both lookups use standard set operations, which are also features of the *Ontology Filter*. The *Concept Lookup* uses the *Union* and *Intersection* set operations, while the *Existential quantification* and *Universal quantification* are utilized by the *Matrix Lookup* operation. If needed, the result set delivered by the *Ontology Filter* can be inversed by the *Not* set operation. This can be helpful in order to determine all products that can be eaten by John instead of all that are forbidden for him, for example.

Ontology concepts can be looked up in the recommender by using the *Concept Lookup*, which needs at least two different matrices for the operation. Hereby, the column domain of the first matrix has to be the row domain of the second matrix. Applied on the first matrix, the *Concept Lookup* filters certain column elements for one single row element based on given constraints. These constraints can be generated by any available *SMART Recommendations Engine* filter, such as the *Existing* or *Feature* filters. The filtered column elements – now selected rows in the row domain of the second matrix – build the basis for further operations. All selected rows will be processed again individually depending on given filters. The calculated result sets of each selected row will be returned and will then be either unified or intersected based on the selected set operation (*Union* or *Intersection*). Figure 11 shows a *Concept Lookup* example, in which ingredients are looked up that are **not** allowed to be eaten by John.

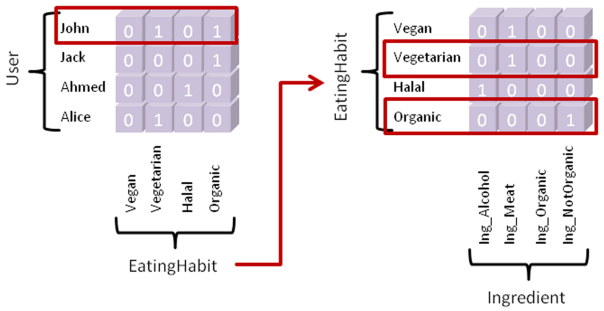


Figure 11: Concept Lookup Example

The *Matrix Lookup* filters information in a matrix based on a given column domain result set of another matrix. Therefore, it also requires the use of two different matrices, whereas the column domain of the first matrix remains the column domain of the second matrix. Rows of the second matrix will be filtered based on the given column domain result set and a predefined set operation (*existential quantification* or *universal quantification*). The result is one set of filtered row elements. In Figure 12, an exemplary *Matrix Lookup* can be seen. All food products are looked up that are **forbidden** for *organic* eating people.

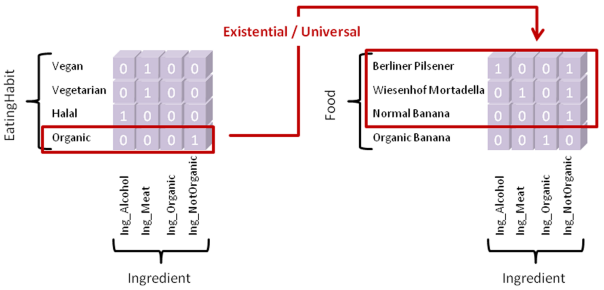


Figure 12: Matrix Lookup Example

An overview of all supported operations of the *Ontology Filter* is given in Table 1.

Complex recommendation queries require combining both lookups to single a *Concept and Matrix Lookup* operation, which is exemplary demonstrated in section 6.

6 Demonstration

Powerful contextual/semantic recommendations can be computed by the recommender using the *SMART Ontology Extension* in combination with the *Proximity Filter*. The ontology data present in the database of the *SMART Recommendations Engine* facilitates the recommendation query

Filter	Matrix Ops.	Set Ops.
Ontology Filter	Concept Lookup	Union
		Intersection
	Matrix Lookup	Existential quantification
		Universal quantification
		Not

Table 1: Ontology Filter Operations

requests of a user by reducing his effort to manually describing his personal needs and interests in a detailed way. In conjunction with the *Proximity Filter*, the recommendations are also filtered by their location leading to a great user experience when using a mobile shopping service, for example. This function is going to be demonstrated in this section by taking the following mobile grocery shopping service scenario as a basis:

Assume that John is vegetarian and eats only organic food products. He wants to buy groceries nearby his location and that's why he wants his shopping application to give him recommendations based on the food categories he prefers. After the selection of food categories for his shopping cart, John gets food products recommended for each category fitting his eating preferences and sorted by their relevance based on John's previous purchases. The grocery stores that sell these products and are in a certain range to John are also listed.

Eating preferences, such as eating vegetarian or organic, include some sort of implicit and indirect knowledge. For example, in order to be able to specify vegetarian preferences, the system must know what kind of food vegetarians do not eat. It further has to have the information, which ingredients are included in each grocery, so that non-vegetarian food can be filtered based on the list of non-consumable ingredients. Since semantic ontologies are predestined to model this kind of dependencies, three ontologies were designed including a complete data set to map relevant information given in the food scenario.

The *food-ingredient-shop* ontology consists of a classification of food categories, concrete grocery products including their ingredients and shops, in which they can be found. The *eatinghabit-ingredient* ontology, on the other hand, models relations between certain eating preferences and the ingredients that are either allowed or forbidden to eat. User profiles are stored in the *user-profile* ontology. All data modeled in these ontologies is exported into the recommender database via the *Ontology Mapping tool*. Figure 13 shows parts of the created data model after a successful export.

Using the *SuQL*, the *SMART Recommendations Engine* can now provide contextual/semantic recommendations based on the available data set and the implemented recommendation algorithms. For the food scenario, assume that John wants to buy snacks and bread in the range of 2 kms from his location and that he already purchased the brown bread product *Naturkind_SonnenblumenVollkornbrot_Geschnitten*. The *SuQL* query is constructed as follows: At first several semantic filterings are performed using the lookup operations several times in order to identify all snack and bread products that fit John's eating preferences and his location. Afterwards, these elements are sorted by their relevance and limited to a certain number depending on the relevance predictions calculated by the recommendation algorithms.

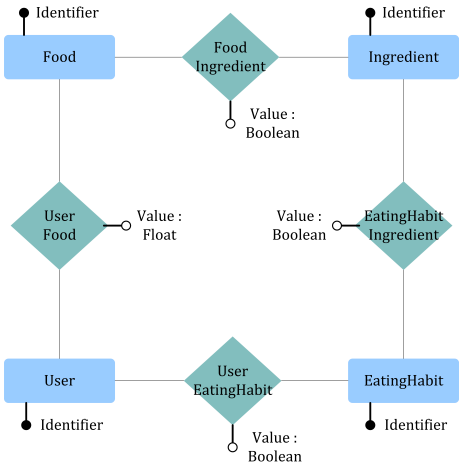


Figure 13: Food Scenario Ontology Data Model

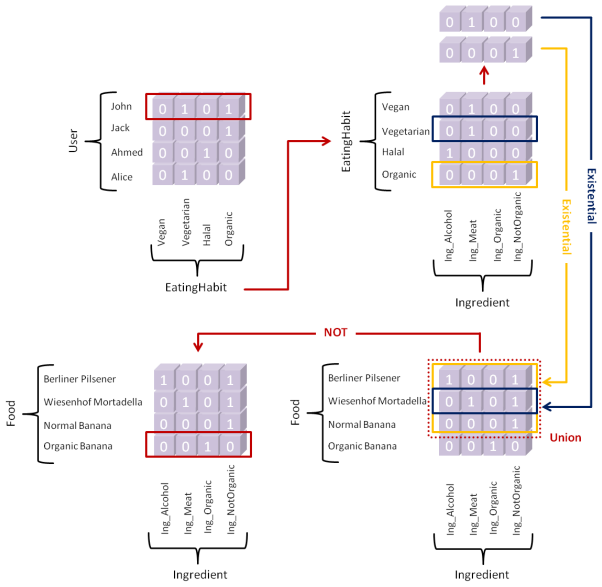


Figure 14: Concept and Matrix Lookup

Figure 14 shows a semantic filtering example for John, in which the *Ontology Filter* first performs a *Concept Lookup* in the *User x EatingHabit* matrix that looks up his eating preferences. In the second matrix (*EatingHabit x Ingredient*), John's eating preferences (*vegetarian* and *organic*) are mapped to the ingredients. (Note: Due to the fact that organic food products cannot be identified by their ingredients, artificial ingredients, such as *Ing_Organic* or *Ing_NotOrganic* were created and related to the food products.) The *Matrix Lookup* then looks up all groceries in the *Food x Ingredient* matrix for vegetarians and organic eating people individually. Finally, both result sets are unified to one single result set and inversed by the *Not* set operation in order to get a set of groceries, which can be eaten by John. These groceries are also filtered by their categories, so that only snacks and bread products remain.

Another semantic operation in form of a *Matrix Lookup* is utilized, so that all grocery stores near to John's location

that sell these food products can be looked up. This is done by using the *Proximity Filter*, where the location of John is specified as well as the range to look for. All grocery stores near to John's position are delivered to the *Matrix Lookup*, which then performs a lookup to find all filtered snacks and bread products that are available in these shops.

In a final step, the recommendation algorithm is used in the recommendation process. Even though the engine can also generate recommendations based on different *collaborative filtering* algorithms, this paper focuses on an extended version of the *content-based filtering* approach using the ontology taxonomy as content meta-data. This approach – named as the *ontology-based filtering* algorithm – calculates relevance predictions using the similarity of food products based on their cagetory as content meta-data (e.g. *brown bread* is more similar to *white bread* than to snacks) and the implicit user feedback given by users automatically when purchasing items. This user feedback affects the final recommendations in that way that groceries of a category the user has already bought products before become more relevant to him than products of other nodes in the tree.

The ontology taxonomy stored in the *Food x Class* matrix is used to compute a similarity between groceries by means of their categories leading to a food similarity matrix with the dimensions *Food x Food*. *User x Food* relevance predictions based on the taxonomy information can be gained by applying a matrix product transformation on the *User x Food* feedback matrix and the *Food x Food* similarity matrix (see Figure 15).

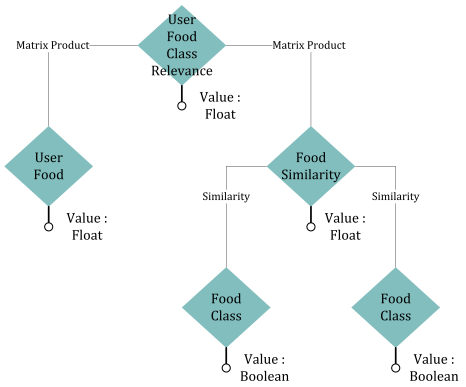


Figure 15: Ontology-based Filtering Algorithm

Figure 16 shows the *SuQL* recommender response to John's query. John bought *Naturkind_SonnenblumenVollkornbrot_Geschnitten* before, which is from the bread product category *BrownBread*. This and all other *BrownBread* products are most relevant to him. The relevance value decreases the more vegetarian and organic bread products are in other nodes of the *Bread* tree. There is only one snack fitting his eating preferences with a low relevance value since John did not purchase any snacks yet. As a result, five bread products and one snack are presented in combination with the grocery stores nearby that sell these products.

7 Discussion of the Approach

One way to compare the presented approach with widespread memory-based and model-based recommendation algorithms is to compare the accuracy of these approaches based on a common metric such as the mean average error (MAE). For that, a suitable set of data points for training and testing is needed. The chosen application domain of the presented demonstration scenario was predetermined by the project's context in which our approach was developed. This context did not provide us with a suitable data set for this kind of practical testing. However, a closer examination of our approach shows that a benchmark recommendation algorithm, which has been adapted to make use of the product hierarchy that is stored in the ontology, provides better relevance values by taking the relation of different food items into account. The relevance values of unrelated items decrease in less relevant or unsuitable nodes of the product hierarchy, so that the overall recommendation quality in terms of the achieved accuracy increases.

Figure 16: Ontology-based *SuQL* Recommender Response

8 Conclusion

This paper introduced the generic recommender system, the *SMART Recommendations Engine* of Fraunhofer FOKUS, which is capable of providing recommendations for different types of applications. In order to make the engine meet the recommendation quality requirements of modern mobile services, the recommender was extended by two new extensions. The *SMART Multidimensionality Extension* provides multidimensionality capabilities to the so far two-dimensional recommender system. This upgrade enables the recommender to deal with additional context dimensions in order to generate more accurate recommendations taking user's current contextual situation into account. The *SMART Ontology Extension*, on the hand, exploits semantic ontology data by exporting all relevant information given in pre-designed application-dependent ontologies into the data structure of the *SMART Recommendations Engine* enabling the recommender to use semantic knowledge or ontology taxonomy information for recommendation purposes.

The demonstration showed that in conjunction with the *Proximity Filter*, the *SMART Ontology Extension* provides an added-value to the engine by generating semantic and contextual recommendations.

References

[Adomavicius and Tuzhilin, 2005] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. on Knowl. and Data Eng.*, 17(6):734–749, 2005.

[Adomavicius *et al.*, 2005] Gediminas Adomavicius, Ramesh Sankaranarayanan, Shahana Sen, and Alexander Tuzhilin. Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Trans. Inf. Syst.*, 23(1):103–145, 2005.

[Chen, 2005] Annie Chen. Context-aware collaborative filtering system: predicting the user's preferences in ubiquitous computing. In *CHI '05: CHI '05 extended abstracts on Human factors in computing systems*, pages 1110–1111, New York, NY, USA, 2005. ACM.

[Costa *et al.*, 2007] A.C. Costa, R.S.S. Guizzardi, G. Guizzardi, and J.G.P. Filho. Cores: Context-aware, ontology-based recommender system for service recommendation. *19th International Conference on Advanced Information Systems Engineering (CAISE07)*, 2007.

[Dean and Schreiber, 2004] M. Dean and G. Schreiber. Owl web ontology language reference, 2004. http://www.w3.org/TR/owl-ref/.

[Farsani and Nematbakhsh, 2006] H.K. Farsani and M. Nematbakhsh. A semantic recommendation procedure for electronic product catalog. *International Journal of Applied Mathematics and Computer Sciences*, 3:86–91, 2006.

[Kim and Kwon, 2007] S. Kim and J. Kwon. Effective context-aware recommendation on the semantic web. *International Journal of Computer Science and Network Security*, 7:154–159, 2007.

[Pereira Filho *et al.*, 2006] J.G. Pereira Filho, R.M. Pessoa, and C.Z. Calvi. Infraware: A support middleware to context-aware mobile applications (in portuguese). *In Proceedings of the 24th Simpsio Brasileiro de Redes de Computadores*, 2006.

[Raeck and Steinert, 2010] Christian Raeck and Fabian Steinert. Fraunhofer institute fokus, smart recommendations engine, 2010. http://tinyurl.com/3xdsffz.

[Setten *et al.*, 2004] Mark Van Setten, Stanislav Pokraev, Johan Koolwaaij, and Telematica Instituut. Context-aware recommendations in the mobile tourist application compass. In *In Nejdl, W. and De Bra, P. (Eds.). AH 2004, LNCS 3137*, pages 235–244. Springer-Verlag, 2004.

[Shenk, 1998] David Shenk. *Data Smog: Surviving the Information Glut*. Harper San Francisco, 1998.

[Yu *et al.*, 2006] Zhiwen Yu, Xingshe Zhou, Daqing Zhang, Chung-Yau Chin, Xiaohang Wang, and Ji Men. Supporting context-aware media recommendations for smart phones. *IEEE Pervasive Computing*, 5(3):68–75, 2006.

1 Author Index