Proceedings of the ECML/PKDD – 2003

First European Web Mining Forum

Edited by
Bettina Berendt, Andreas Hotho, Dunja Mladenic,
Maarten van Someren, Myra Spiliopoulou, and Gerd Stumme

14th European Conference on Machine Learning
7th European Conference on Principles and Practice of Knowledge Discovery in Databases

22 – 26 September 2003
Cavtat - Dubrovnik
Croatia

ISBN: 953-6690-31-4

Proceedings of the ECML/PKDD – 2003
First European Web Mining Forum

Edited by
Bettina Berendt, Andreas Hotho, Dunja Mladenic,
Maarten van Someren, Myra Spiliopoulou, and Gerd Stumme

Workshop Chairs: Stefan Kramer, Technische Universität München
Luis Torgo, University of Porto

Publisher: Ruđer Bošković Institute, Bijenička cesta 54, P.O.B. 180, 10002 Zagreb, Croatia

ISBN: 953-6690-31-4

Zagreb 2003

ECML/PKDD-2003: 1st European Web Mining Forum [EWMF 2003]

The Web presents a key driving force for a large spectrum of applications in which a user interacts with a company, a governmental authority, a non-governmental organization, or another non-profit institution. The application providers should combine knowledge about user expectations and Web semantics, in order to form personalised, user-friendly, and business-optimal services. Enabling methodologies include data mining, text mining, and ontology learning. Recipient technologies include user profiling, usage analysis, ontology extraction for the Semantic Web, intelligent search and recommendation systems based on user preferences, page content, and site semantics.

The EWMF'03 workshop is organised by the KDNet interest group Web Mining Forum on knowledge discovery from and about the Web. It aims to bring together various perspectives on Web mining and stress the synergy effects between Web usage mining, Web content mining and Web intelligence, and of Semantic Web Mining.

We would like to take this opportunity to thank the members of the Program Committee and the secondary reviewers for their great assistance in reviewing the submitted publications.

By Bettina Berendt, Andreas Hotho, Dunja Mladenic, Maarten van Someren, Myra Spiliopoulou, Gerd Stumme, July 2003

Organizing Committee

- Bettina Berendt
  Institute of Information Systems, Humboldt University Berlin,
  email: berendt@wiwi.hu-berlin.de
- Andreas Hotho
  Institute AIFB, Universität Karlsruhe, Germany
  email: hotho@aifb.uni-karlsruhe.de
- Dunja Mladenic
  J. Stefan Institute, Department for Intelligent Systems, Ljubljana, Slovenia
  email: Dunja.Mladenic@ijs.si
- Maarten van Someren
  SWI, University of Amsterdam, The Netherlands
  email: maarten@swi.psy.uva.nl
- Myra Spiliopoulou
  Institute of Technical and Business Information Systems, Otto-von-Guericke-Universitaet Magdeburg
  email: myra@iti.cs.uni-magdeburg.de
- Gerd Stumme
  Institute AIFB, Universität Karlsruhe, Germany
  email: stumme@aifb.uni-karlsruhe.de

Program Committee

- Ed Chi (Xerox Parc, Palo Alto, USA)
- Ronen Feldman (Bar-Ilan University, Ramat Gan, Israel)
- Marko Grobelnik (J. Stefan Institute, Ljubljana, Slovenia)
- Oliver Günther (Humboldt University Berlin, Germany)
- Stefan Haustein (University of Dortmund, Germany)
- Jörg-Uwe Kietz (kdlabs AG, Zuerich, Switzerland)
- Ee-Peng Lim (Nanyang Technological University, Singapore)
- Alexander Maedche (Robert Bosch GmbH, Stuttgart, Germany)
- Brij Masand (Data Miners, Boston, USA)
- Yannis Manolopoulos (Aristotle University, Greece)
- Ryszard S. Michalski (George Mason University, USA)
- Bamshad Mobasher (DePaul University, Chicago, USA)
- Claire Nedellec (Université Paris Sud, France)
- George Paliouras (Demokritos National Centre for Scientific Research, Athens, Greece)
- Jian Pei (Simon Fraser University, Canada)
- John R. Punin (Rensselaer Polytechnic Institute, Troy, N.Y., USA)
- Jaideep Srivastava (University of Minnesota, Minneapolis, USA)
- Rudi Studer (University of Karlsruhe, Germany)
- Stefan Wrobel (Fraunhofer Institute for Autonomous Intelligent Systems, Sankt Augustin, Germany)
- Mohammed Zaki (Rensellaer Polytechnic Institute, USA)
- Osmar Zaiane (University of Alberta, Edmonton, Alberta / Canada)

Secondary reviewers

- Steffan Baron (Humboldt University Berlin, Germany)
- Prasanna Desikan (University of Minnesota, Minneapolis, USA)
- Haiyang Liu (University of Minnesota, Minneapolis, USA)
- Nenad Stojanovic (University of Karlsruhe, Germany)

This Workshop is partially supported by the European FP5 FET project "European Knowledge Discovery Network of Excellence (KDNet)".

Table of contents

Mining Web sites using wrapper induction, named entities and post-processing

Georgios Sigletos[1,2], Georgios Paliouras[1],
Constantine D. Spyropoulos[1], Michalis Hatzopoulos[2]

[1] Institute of Informatics and Telecommunications, NCSR "Demokritos",
P.O. BOX 60228, Aghia Paraskeyh, GR-153 10, Athens, Greece
{sigletos, paliourg, costass}@iit.demokritos.gr
[2] Department of Informatics and Telecommunications, University of Athens,
TYPA Buildings, Panepistimiopolis, Athens, Greece
{sigletos, mike}@di.uoa.gr

Abstract. This paper presents a novel method for extracting information from collections of Web pages across different sites. Our method uses a standard wrapper induction algorithm and exploits named entity information. We introduce the idea of post-processing the extraction results for resolving ambiguous facts and improve the overall extraction performance. Post-processing involves the exploitation of two additional sources of information: fact transition probabilities, based on a trained bigram model, and confidence probabilities, estimated for each fact by the wrapper induction system. A multiplicative model that is based on the product of those two probabilities is also considered for post-processing. Experiments were conducted on pages describing laptop products, collected from many different sites and in four different languages. The results highlight the effectiveness of our approach.

1 Introduction

Wrapper induction (WI) [7] aims to generate extraction rules, called *wrappers*, by mining highly structured collections of Web pages that are labeled with domain-specific information. At run-time, wrappers extract information from unseen collections and fill the slots of a predefined template. These collections are typically built by querying an appropriate search form in a Web site and collecting the response pages, which commonly share the same content format.

A central challenge to the WI community is *Information Extraction* (IE) from pages across multiple sites, including unseen sites, by a single trained system. Pages collected from different sites usually exhibit multiple hypertext markup structures, including tables, nested tables, lists, etc. Current WI research relies on learning separate wrappers for different structures. Training an effective site-independent IE system is an attractive solution in terms of *scalability*, since any domain-specific page could be processed, without relying heavily on the hypertext structure.

In this paper we present a novel approach to IE from Web pages across different sites. The proposed method relies on domain specific *named entities*, identified within

Web pages. Those entities are embedded within the Web pages as XML tags and can serve as a page-independent common markup structure among pages from different sites. A standard WI system can be applied and exploit the additional textual information. Thus, the new system relies more on page-independent named-entity markup tags for inducing delimiter-based rules for IE and less on the hypertext markup tags, which vary among pages from multiple sites.

We experimented with STALKER [10], which performs extraction from a wide range of Web pages, by employing a special formalism that allows the specification of the output multi-place schema for the extraction task. However, information extraction from pages across different sites is a very hard problem, due to the multiple markup structures that cannot be described by a single formalism. In this paper we suggest the use of STALKER for *single-slot* extraction, i.e. extraction of isolated *facts* (i.e. *extraction fields*), from pages across different sites.

A further contribution of this paper is a method for post-processing the system's extraction results in order to disambiguate facts. When applying a set of single-slot extraction rules to a Web page, one cannot exclude the possibility of identical or overlapping textual matches within the page, among different rules. For instance, rules for extracting instances of the facts *cd-rom* and *dvd-rom* in pages describing laptop products may overlap or exactly match in certain text fragments, resulting in ambiguous facts. Among these facts, the correct choice must be made.

To deal with the issue of ambiguous facts, two sources of information are explored: *transitions* between facts, incorporated in a bigram model, and *prediction confidence* values, generated by the WI system. Deciding upon the correct fact can be based on information from either the trained bigram model and/or the confidence assigned to each predicted fact. A multiplicative model that combines these two sources of information is also presented and compared to each of the two components.

The rest of this paper is structured as follows: In Section 2 we outline the architecture of our approach. Section 2.1 briefly describes the named entity recognition task. Section 2.2 reviews STALKER and in Section 2.3 we discuss how STALKER can be used under the proposed approach to perform IE from pages across different sites. In Section 3 we discuss the issue of post-processing the output of the STALKER system in order to resolve ambiguous facts. Section 4 presents experimental results on datasets that will soon be publicly released. Related work is presented in Section 5. Finally we conclude in Section 6, discussing potential improvements of our approach.

2 Information Extraction from multiple Web sites

Our methodology for IE from multiple Web sites is graphically depicted in Figure 1. Three component modules process each Web page. First, a *named-entity recognizer* (NER) that identifies domain-specific named entities across pages from multiple sites. A trained WI system is then applied to perform extraction. Finally, the extraction results are post-processed to improve the extraction performance.

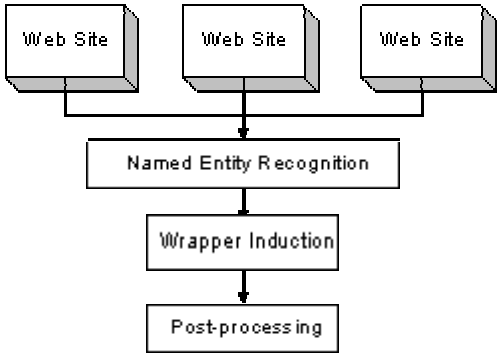


Fig. 1. Generic architecture for information extraction from multiple Web sites

2.1 Named entity recognition

Named entity recognition (NER) is an important subtask in most language engineering applications and has been included as such in all MUC competitions, e.g. [8]. NER is best known as the first step in the IE task, and involves the identification of a set of basic *entity names*, *numerical expressions*, *temporal expressions*, etc. The overall IE task aims to extract *facts* in the form of multi-place relations and NER provides the entities that fill the argument slots. NER has not received much attention in Web IE tasks.

We use NER in order to identify basic named entities relevant to our task and thus reduce the complexity of fact extraction. The identified entities are identified within Web pages as XML tags and serve as a valuable source of information for the WI system that follows and extracts the facts. Some of the entity names (*ne*), numerical (*numex*) and temporal (*timex*) expressions, used in the laptop domain are shown in Table 1, along with the corresponding examples of XML tags.

Table 1. Subset of named entities for the laptop domain

Entity	*Entity Type*	*Examples of XML tags*
Ne	Model, Processor	<*ne type=Model*>Presario</*ne*> <*ne type=Processor*>Intel Pentium </*ne*>
Numex	Capacity, Speed	<*numex type=Speed*>300 MHz </*numex*> <*numex type=Capacity*>20 GB </*numex*>
Timex	Duration	<*timex type=Duration*>1 year </timex>

2.2 The STALKER wrapper induction system

STALKER [10] is a sequential covering rule learning system that performs *single-slot* extraction from highly-structured Web pages. *Multi-slot* extraction –i.e. linking of the

isolated facts- is feasible through an *Embedded-Catalog (EC) Tree* formalism, which may describe the common structure of a range of Web pages. The EC tree is constructed manually, usually for each site, and its leaves represent the individual facts. STALKER is capable of extracting information from pages with tabular organization of their content, as well as pages with hierarchically organized content.

Each extraction rule in STALKER consists of two ordered lists of *linear landmark automata* (LA's), which are a subclass of nondeterministic finite state automata. The first list constitutes the *start* rule, while the second list constitutes the *end* rule. Using the EC tree as a guide, the extraction in a given page is performed by applying –for each fact- the LA's that constitute the start rule in the order in which they appear in the list. As soon as a LA is found that matches within the page, the matching process terminates. The process is symmetric for the end rule. More details on the algorithm can be found in [10].

2.3 Adapting STALKER to multi-site information extraction

The EC tree formalism used in STALKER is generally not applicable for describing pages with variable markup structure. Different EC trees need to be manually built for different markup structures and thus different extraction rules to be induced. In this paper, we are seeking for a single domain-specific trainable system, without having to deal with each page structure separately. The paper focuses on the widely-used approach of single-slot extraction. Our motivation is that if isolated facts could be accurately identified, then it is possible to link those facts separately on a second step. We therefore specify our task as follows:

For each fact, try to induce a *list iteration* rule as depicted in Figure 2.

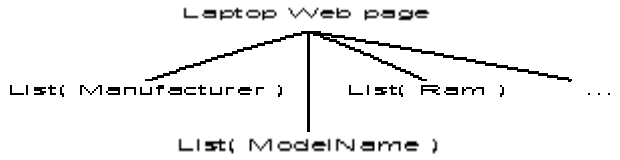


Fig. 2. Simplification of the EC tree. A list iteration rule is learned for each fact and applies to the whole content of a page, at run-time

The EC tree depicted in Figure 2 has the following interpretation: a Web page that describes laptop products consists of a list of instances of the fact *Manufacturer* (e.g. "Compaq"), a list of instances of the fact *ModelName* (e.g. "Presario"), a list of *Ram* instances (e.g. "256MB"), etc. The system, during runtime, exhaustively applies each rule to the content of the whole page. This simplified EC tree is independent of any particular page structure. The proposed approach relies on the page-independent named entities to lead to efficient extraction rules.

Since each extraction rule applies exhaustively within the complete Web page, rather than being constrained by the EC tree, we expect an extraction bias towards recall, i.e., overgeneration of extracts for each fact. The penalty is a potential loss in precision, since each rule applies to text regions that do not contain relevant

information and may return erroneous instances. Therefore we seek a post-processing mechanism capable of discarding the erroneous instances and thus improving the overall precision.

3 Post-processing the extraction results

In single-slot IE systems, each rule is applied independently of the others. This may naturally cause identical or overlapping matches among different rules resulting in multiple ambiguous facts for those matches. We would like to resolve such ambiguities and choose the correct fact. Choosing the correct fact and removing all the others shall improve the extraction precision.

3.1 Problem specification

In this paper we adopt a post-processing approach in order to resolve ambiguities in the extraction results of the IE system. More formally, the task can be described as follows:

1. Let D be the sequence of a document's tokens and T_j (s_j, e_j) a fragment of that sequence, where s_j and e_j are the start and end token bounds respectively.
2. Let I = $\{i_j \mid i_j: T_j \rightarrow fact_j\}$ be the set of instances extracted by all the rules, where $fact_j$ is the predicted fact associated with instance T_j.
3. Let DT be the list of all *distinct* text fragments T_j, appearing in the extracted instances in I. Note that $T_1(s_1, e_1)$ and $T_2(s_2, e_2)$ are different, if either $s_1 \neq s_2$ or $e_1 \neq e_2$. The elements of DT are sorted in ascending order of s_j.
4. If for a distinct fragment T_i in DT, there exist at least two instances i_k and i_l so that $i_k: T_i \rightarrow fact_k$ and $i_l : T_i \rightarrow fact_l$, $k \neq l$, then $fact_k$ and $fact_l$ are ambiguous facts for T_i.
5. The goal is to associate a single fact to each element of the list DT.

To illustrate the problem, if for the fragment $T_j(24, 25)$="16 x" in a page describing laptops, there are two extracted instances i_k and i_l, where $fact_k$ = *dvdSpeed* and $fact_l$ = *cdromSpeed*, then there are two ambiguous facts for T_i. One of them must be chosen and associated with T_j.

3.2 Formulate the task as a hill-climbing search

Resolving ambiguous facts can be viewed as a *hill-climbing* search in the space of all possible sequences of facts that can be associated with the sequence DT of distinct text fragments.

This hill-climbing search can be formulated as follows:

1. Start from a hypothetical empty node, and transition at each step *j* to the next distinct text fragment T_j of the sorted sequence DT.

2. At each step apply a set of operations *Choose* ($fact_k$). Each operation associates T_j with the $fact_k$ predicted by an instance $i_k = \{T_j \rightarrow fact_k\}$. A *weight* is assigned to each operation, based on some predefined metric. The operation with the highest weight is selected at each step.
3. The goal of the search is to associate a single fact to the last distinct fragment of the sorted list DT, and thus return the final unambiguous sequence of facts for DT.

To illustrate the procedure, consider the fictitious token table in Table 2(a), which is part of a page describing laptop products.

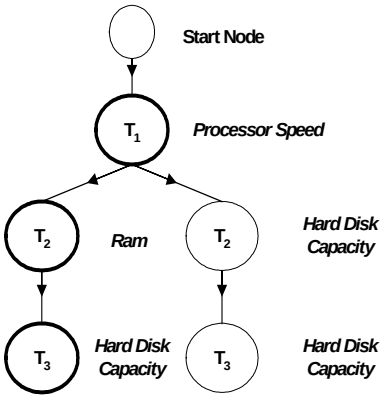
Table 2. (a) Part of a token table of a page dscribing laptops. (b) Instances extracted by STALKER. (c) The tree of all possible fact paths (d) The extracted instances after the disambiguation process

…	*33*	*34*	*35*	*36*	*37*	*38*	*39*	*40*	…
…	1,6	GHz	/	1	GB	/	80	GB	…

(a)

T_j (s_j, e_j)	$fact_k$
T_1 (33, 34)	Processor Speed
T_2 (36, 37)	Ram
T_2 (36, 37)	Hard Disk capacity
T_3 (39, 40)	Hard Disk Capacity

(b)



(c)

T_j (s_j, e_j)	$fact_k$
T_1 (33, 34)	Processor Speed
T_2 (36, 37)	Ram
T_3 (39, 40)	Hard Disk Capacity

(d)

Table 2(b) lists the instances extracted by STALKER for the token table part of Table 2(a). The DT list consists of the three distinct fragments T_1, T_2, T_3. Table 2(c) shows the two possible fact sequences that can be associated with DT. After the *processor speed* fact prediction for T_1, two operations apply for predicting a fact for T_2: The *choose (Ram)* and *choose (hard disk capacity)* operations, each associated with a *weight*, according to a predefined metric. We assume that the former operation returns a higher weight value and therefore *Ram* is the chosen fact for T_2. The bold circles in the tree show the chosen sequence of facts {*Processor speed*, *Ram*, *Hard disk capacity*} that is attached to the sequence T_1 T_2 T_3. Table 2(d) illustrates the final extracted instances, after the disambiguation process.

In this paper we explore three metrics for assigning weights to the choice operations:

1. *Confidence* values, estimated for each fact by the WI algorithm.

2. *Fact-transition* probabilities, learned by a bigram model.
3. The *product* of the above probabilities, based on a simple multiplicative model.

Selecting the correct instance, and thus the correct fact, at each step and discarding the others, results in improving the overall precision. However, an incorrect choice harms both the recall and the precision of a certain fact. The overall goal of the disambiguation process is to improve the overall precision while keeping recall unaffected.

3.3 Estimating fact confidence

The original STALKER algorithm does not assign confidence values to the extracted instances. In this paper we estimate confidence values by calculating a value for each extraction rule, i.e. for each fact. That value is calculated as the average *precision* obtained by a three-fold cross-validation methodology on the training set. According to this methodology, the training data is split into three equally-sized subsets and the learning algorithm is run three times. Each time two of the three pieces are used for training and the third is kept as unseen data for the evaluation of the induced extraction rules. Each of the three pieces acts as the evaluation set in one of the three runs and the final result is the average over the three runs.

At runtime, each instance extracted by a single-slot rule will be assigned the *precision* value of that rule. For example, if the text fragment "300 Mhz" was matched by the *processor speed* rule, then this fragment will be assigned the confidence associated with *processor speed*. The key insight into using confidence values is that among ambiguous facts, we can choose the one with the highest estimated confidence.

3.4 Learning fact transition probabilities

In many extraction domains, some facts appear in an almost fixed order within each page. For instance, a page describing laptop products may contain instances of the *processor speed* fact, appearing almost immediately after instances of the *processor name* fact. Training a simple bigram model is a natural way of modeling such dependencies and can be easily implemented by calculating ratios of counts (maximum likelihood estimation) in the labeled data as follows:

$$P(i \to j) = \frac{c(i \to j)}{\sum_{j \in K} c(i \to j)}, \quad \textbf{(1)}$$

where the nominator counts the transitions from fact *i* to fact *j*, according to the labeled training instances. The denominator counts the total number of transitions from fact *i* to all facts (including self-transitions). We also calculate a *starting*

probability for each fact, i.e. the probability that an instance of a particular fact is the first one appearing in the labeled training pages.

The motivation for using fact transitions is that between ambiguous facts we could choose the one with the highest transition probability given the preceding fact prediction. To illustrate that, consider that the text fragment "16 x" has been identified as both *cdromSpeed* and *dvdSpeed* within a page describing laptops. Assume also that the preceding fact prediction of the system is *ram*. If the transition from *ram* to *dvdSpeed* has a higher probability, according to the learned bigram, than from *ram* to *cdromSpeed*, then we can choose the *dvdSpeed* fact. If ambiguity occurs at the first extracted instance, where there is no preceding fact prediction available, then we can choose the fact with the highest starting probability.

3.5 Employing a multiplicative model

A simple way to combine the two sources of information described above is through a multiplicative model, assigning a confidence value to each extracted instance $i_k : T_i \rightarrow fact_k$, based on the *product* of the confidence value estimated for $fact_k$ and the transition probability from the preceding instance to $fact_k$. Using the example of Table 2 with the two ambiguous facts *ram* and *hard disk capacity* for the text fragment T_2, Table 4 depicts the probabilities assigned to each fact by the two methods described in sections 3.3 and 3.4 and the multiplicative model.

Table 3. Probabilities assigned to each of the two ambiguous facts of the text fragment T_2 of Table 2

T_2 (36, 37) = "1 GB"	*WI-Confidence*	*Bigram*	*Multiplicative*
Ram	0,7	0,3	0,21
Hard disk capacity	0,4	0,5	0,20

Using the WI confidence values, the *ram* is selected. However, using bigram probabilities, the *hard disk capacity* is selected. We also experimented with a model that averages the two probabilities, rather than multiplying them. However the experiments led to worse results.

4 Experiments

4.1 Dataset description

Experiments were conducted on four language corpora (Greek, French, English, Italian) describing laptop products. The corpora were collected in the context of CROSSMARC[1].

Approximately 100 pages from each language were hand-tagged using a Web page annotation tool [14]. The corpus for each language was divided into two equally sized data sets for *training* and *testing*. Part of the test corpus was collected from sites not appearing in the training data. The named entities were embedded as XML tags within the pages of the training and test data, as illustrated in Table 1. A separate NER module was developed for each of the four languages of the project.

A total of 19 facts were hand-tagged for the laptop product domain. The pages were collected from multiple vendor sites and demonstrate a rich variety of structure, including tables, lists etc. Examples of facts are the name of the manufacturer, the name of the model, the name of the processor, the speed of the processor, the ram, etc.

4.2 Results

Our goal was to evaluate the effect of named entity information to the extraction performance of STALKER and compare the three different methods for resolving ambiguous facts.

We, therefore, conducted two groups of experiments. In the first group we evaluated STALKER on the testing datasets for each language, with the named entities embedded as XML tags within the pages. Table 4 presents the results. The evaluation metrics are *micro-average recall* and *micro-average precision* [12] over all 19 facts. The last row of Table 4 averages the results over all languages.

Table 4. Evaluation results for STALKER in four languages

Language	*Micro Precision (%)*	*Micro Recall (%)*
Greek	60,5	86,8
French	64,1	93,7
English	52,2	85,1
Italian	72,8	91,9
Average	**62,4**	**89,4**

The exhaustive application of each extraction rule to the whole content of a page resulted, as expected, in a high recall, accompanied by a lower precision. However, named-entity information led a pure WI system like STALKER to achieve a bareable

[1] http://www.iit.demokritos.gr/skel/crossmarc. Datasets will soon be available on this site.

level of extraction performance across pages with variable structure. We also trained STALKER on the same data without embedding named entities within the pages. The result was an unacceptably high training time, accompanied by rules with many disjuncts that mostly overfit the training data. Evaluation results on the testing corpora provided recall and precision figures below 30%.

In the second group of experiments, we evaluated the post-processing methodology for resolving ambiguous facts that was described in Section 3. Results are illustrated in Table 5.

Table 5. Evaluation results after resolving ambiguities

Language	*Micro Precision (%)*			*Micro Recall (%)*		
	WI-Conf.	*Bigram*	*Mult.*	*WI-Conf.*	*Bigram*	*Mult.*
Greek	69,3	73,5	73,8	76,9	81,6	81,9
French	77,0	78,9	79,4	82,1	84,1	84,6
English	65,9	67,5	68,9	74,4	76,2	77,5
Italian	84,4	83,8	84,4	87,6	87,0	87,6
Average	**74,2**	**75,9**	**76,6**	**80,3**	**82,2**	**82,9**

Comparing the results of Table 4 to the results of Table 5, we conclude the following:

1. Choosing among ambiguous facts, using any of the three methods, achieves an overall increase in precision, accompanied by a lower decrease in recall. Results are very encouraging, given the difficulty of the task.
2. Using bigram fact transitions for choosing among ambiguous facts achieves better results that using confidence values. However, the simple multiplicative model outperforms slightly the two single methods.

To corroborate the effectiveness of the multiplicative model, we counted the number of correct choices made by the three post-processing methods at each step of the hill-climbing process, as described in section 3.2. Results are illustrated in Table 6.

Table 6. Counting the ambiguous predictions and the correct choices

Language	*Distinct* T_i	*Ambiguous* T_i	*Corrected (WI-Conf.)*	*Corrected (Bigram)*	*Corrected (Mult.)*
Greek	549	490	251	331	336
French	720	574	321	364	374
English	2203	1806	915	996	1062
Italian	727	670	538	458	483
Average	**1050**	**885**	**506**	**537**	**563**

The first column of Table 6 is the number of distinct text fragments T_i, as defined in section 3.1, for all pages in the testing corpus. The second column counts the T_i with more than one –ambiguous- facts (e.g. the T_2 in Table 2). The last three columns count the correct choices made by each of the three methods.

We conclude that by using a simple multiplicative model, based on the product of bigram probabilities and STALKER-assigned confidence probabilities we make more correct choices than by using either of the two methods individually.

5 Related Work

Extracting information from multiple Web sites is a challenging issue for the WI community. Cohen and Fun [3] present a method for learning page-independent heuristics for IE from Web pages. However they require as input a set of existing wrappers along with the pages they correctly wrap. Cohen *et al.* [4], also present one component of a larger system that extracts information from multiple sites. A common characteristic of both the aforementioned approaches is that they need to encounter separately each different markup structure during training. In contrast to this approach, we examine the viability of trainable systems that can generalize over unseen sites, without encountering each page's specific structure.

An IE system that exploits shallow linguistic pre-processing information is presented in [2]. However, they generalize extraction rules relying on lexical units (tokens), each one associated with shallow linguistic information, e.g., lemma, part-of-speech tag, etc. We generalize rules relying on named entities, which involve contiguous lexical units, and thus providing higher flexibility to the WI algorithm.

An ontology-driven IE system from pages across different sites is presented in [5]. However, they rely on hand-crafted (provided by an ontology) regular expressions, along with a set of heuristics, in order to identify single-slot facts within a document. On the other hand, we try to induce such expressions using wrapper induction.

All systems mentioned in this section experiment with different corpora, and thus cannot easily be comparatively evaluated.

6 Conclusions and Future Work

This paper presented a methodology for extracting information from Web pages across different sites, which is based on using a pipeline of three component modules: a named-entity recognizer, a standard wrapper induction system, and a post-processing module for disambiguating extracted facts. Experimental results showed the viability of our approach.

The issue of disambiguating facts is important for single-slot IE systems used on the Web. For instance, *Hidden Markov Models* (HMMs) [11] are a well-known learning method for performing single-slot extraction [6], [13]. According to this approach, a single HMM is trained for each fact. At run-time, each HMM is applied to a page, using the *Viterbi* procedure, to identify relevant matches. Identified matches across different HMMs may be identical or overlapping resulting in ambiguous facts. Our post-processing methodology can thus be particularly useful to HMM extraction tasks.

Bigram modeling is a simplistic approach to the exploitation of dependencies among facts. We plan to explore higher-level interdependencies among facts, using higher order n-gram models, or probabilistic FSA, e.g. as learned by the *Alergia* algorithm [1]. Our aim is to further increase the number of correct choices made for ambiguous facts, thus further improving both recall and precision. Dependencies among facts shall be also investigated in the context of *multi-slot* extraction.

A bottleneck in existing approaches for IE is the labeling process. Despite the use of a user-friendly annotation tool [14], the labeling process is a tedious, time-consuming and error-prone task, especially when moving to a new domain. We plan to investigate *active learning* techniques [9] for reducing the amount of labeled data required. On the other hand, we anticipate that our labeled datasets will be of use as benchmarks for the comparative evaluation of other current and/or future IE systems.

References

1. Carrasco, R., Oncina, J., Learning stochastic regular grammars by means of a state-merging method. *Grammatical Inference and Applications, ICGI'94*, p. 139-150, Spain (1994).
2. Ciravegna, F., Adaptive Information Extraction from Text by Rule Induction and Generalization. *In Proceedings of the 17th IJCAI Conference.* Seattle (2001).
3. Cohen, W., Fan, W., Learning page-independent heuristics for extracting data from Web pages. *In the Proceedings of the 8th international WWW conference (WWW-99).* Toronto, Canada (1999).
4. Cohen, W., Hurst, M., Jensen, L., A Flexible Learning System for Wrapping Tables and Lists in HTML Documents. *Proceedings of the 11th International WWW Conference.* Hawaii, USA (2002).
5. Davulcu, H., Mukherjee, S., Ramakrishman, I.V., Extraction Techniques for Mining Services from Web Sources, *IEEE International Conference on Data Mining*, Maebashi City, Japan (2002).
6. Freitag, D., McCallum, A.K., Information Extraction using HMMs and Shrinkage. *AAAI-99 Workshop on Machine Learning for Information Extraction*, p.31-36 (1999).
7. Kushmerick N., Wrapper induction for Information Extraction, *PhD Thesis, Department Of computer Scienc, Univ. Of Washington* (1997).
8. MUC-7, *http://www.itl.nist.gov/iaui/894.02/related_pr ojects/muc*.
9. Muslea, I., Active Learning with multiple views. *PhD Thesis*, University of Southern California (2002).
10. Muslea, I., Minton, S., Knoblock, C., Hierarchical Wrapper Induction for Semistructured Information Sources. *Journal Of Autonomous Agents and Multi-Agent Systems*, 4:93-114 (2001).
11. Rabiner, L., A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE* 77-2 (1989).
12. Sebastiani F., Machine Learning in Automated Text Categorization, *ACM Computing Surveys*, 34(1):1-47 (2002).
13. Seymore, K., McCallum, A.K., Rosenfeld, R., Learning hidden Markov model structure for information extraction. *AAAI Workshop on Machine Learning for Information Extraction*, p. 37-42 (1999).
14. Sigletos, G., Farmakiotou, D., Stamatakis, K., Paliouras, G., Karkaletsis V., Annotating Web pages for the needs of Web Information Extraction Applications. *Poster at WWW 2003*, Budapest Hungary, May 20-24 (2003).

Semantic Outlier Analysis for Sessionizing Web Logs

Jason J. Jung[1] and Geun-Sik Jo[1]

Intelligent E-Commerce Systems Laboratory,
School of Computer Engineering, Inha University,
253 Yonghyun-dong, Incheon, Korea 402-751
j2jung@intelligent.pe.kr, gsjo@inha.ac.kr

Abstract. As the web usage patterns from clients are getting more complex, simple sessionizations based on time and navigation-oriented heuristics have been restricted to exploit various kinds of rule discovering methods. In this paper, we present semantic session reconstruction based on semantic outliers from web log data. Above all, web directory service such as Yahoo is applied to enrich semantics to web logs, as categorizing them to all possible hierarchical paths. In order to detect the candidate set of session identifiers, semantic factors like semantic mean, deviation, and distance matrix are established. Eventually, each semantic session is obtained based on nested repetition of top-down partitioning and evaluation process. For experiment, we applied this ontology-oriented heuristics to sessionize the access log files for one week from IRCache. Compared with time-oriented heuristics, more than 48% of sessions were additionally detected by semantic outlier analysis. It means that we can conceptually track the behavior of users tending to easily change their intentions and interests, or simultaneously try to search various kinds of information on the web.

1 Introduction

As the concern for searching relevant information from the web has been exponentially increasing, the very large amount of log data have been generated in web servers. There are several types of web logs such as server access logs, referrer logs, agent logs, and client-side cookies. Thus, many applications have been focusing on various ways to analyze them in order to recognize the usage patterns of users and discover other meaningful patterns [2]. For example, on-line newspaper on the web [6], web caching [7], and supporting user web browsing [8], [20] can be told as the domains relevant to analyzing web log data.

In this paper, among the whole steps of web user profiling mentioned in [3], we have taken the session identification for segmenting web log data in consideration. There are mainly two kinds of sessionization heuristics for partitioning each user activity into sequences of entries corresponding to each user visit. First, time-oriented heuristics consider temporal boundaries such as a maximum session length or maximum time allowable for each pageview [4]. Second,

navigation-oriented heuristics such as HITS, PageRank, and DirectHit take the linkage between pages into account [5].

However, knowledge that is extractable from sessions identified by those heuristics is limited like frequent and sequential patterns represented by URLs. It means that web logs has to be sessionized with semantic enrichment based on ontology in order to find out more potential and meaningful information like a user's preference and intention. As shown in Fig. 1, blocks are requests from a user and their shapes are means the concepts related with requested URLs.

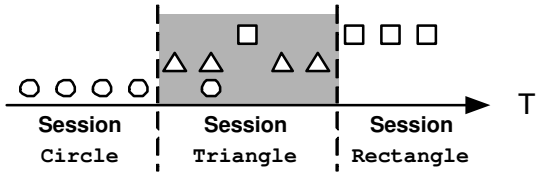


Fig. 1. Streaming Logs

More importantly, web caching(or proxy) servers have to track streaming URL requests from multiple clients, because they have to increase predictability for prefetching web content that is expected in next request.

Enriching web logs with their corresponding semantic information has been attempted in some studies [9], [20]. Typically, there are two kinds of approaches which are the *information extraction* and *information dimensions* mapping URLs to set of concepts as a feature vector and a specific value, respectively. While both of them apply keywords extracted from an URL's web page to perform semantic enrichment, we present conceptualizing an URL information itself by using web directory and introduce representing conceptualized URLs as tree-like information.

In this paper, we describe the conceptualization of web logs based on web directory, and then sessionize them by detecting semantic outliers. Thereby, in the following section, general types of web logs and their data models will be introduced. Sect. 3 will address how to handle url conceptualization against problems of web directory (Sect. 3.1) and how to determine semantic relationships between URLs for detecting semantic outliers (Sect. 3.3). We show semantic sessionization of the streaming web logs and verify our ontology-oriented heuristics approach in the Sect. 4 and Sect. 5, respectively. Finally, Sect. 6 makes a conclusion of this study and mention future work.

2 Data Model of Web Log and Problem Statement

There are several standard data models of web logs such as NCSA, W3C extended, and IRCache[1] format. In this paper we have focused on the IRCache format, which is the NLANR web caching project [1].

Table 1. The Data Format of IRCache Log Files

Fields	Description
Timestamp	The time when the client socket is closed. (millisecond)
Elapsed Time	The elapsed time of the request, in milliseconds.
Client Address	A random IP address identifying the client.
Log Tag	The Log Tag describing how the request was treated locally (hit, miss, etc)
HTTP Code	The HTTP status code
Size	The number of bytes written to the client.
Request Method	The HTTP request method.
URL	The requested URL.
User Ident	Always '-' for the IRCache logs.
Hierarchy Data and Hostname	A description of how and where the requested and Hostname object was fetched.
Content Type	The Content-type field from the HTTP reply.

Each IRCache log file consists of ten basic fields, as shown in the Table 1, and for instance, some records of a log file are as follows.

1048118411.784 214655 165.246.31.128 TCP_MISS/503 1568 GET http://www.intelligent.pe.kr/a.html - NONE/- text/html

1048118421.159 238 218.53.200.251 TCP_MISS/304 261 GET http://www.antara.co.id/images/spacer.gif - DIRECT/202.155.27.190 -

There are, generally, some problems to analyze these web logs such as their anonymity, rotating IP addresses connections through dynamic assignment of ISPs, missing references due to caching, and inability of servers to distinguish among different visits. Therefore, we note the problem statements concentrated for semantic sessionization in this paper, as follows.

- **Weakness of IP address field as session identifier.** The same IP address field in a web logs (within the time window or not) can not guarantee that those requests are caused by only one user, and reversely, requests from the different IPs can be generated by a particular user. As independent of temporal difference, the sequential or contiguous logs whose IP addresses are equivalent can be more partitioned or more agglomerated.
- **Simultaneous user requests based on multiple intention.** An user can request more than an URL "at the same time" by using more than a

[1] This project was administered by the University of California San Diego, in cooperation with the San Diego Supercomputer Center and the National Laboratory for Applied Network Research.

web browser. It means we have to consider multiple intention of users by classifying mixed logs according to the corresponding semantics.

Each request consists of timestamp, IP address, and URL fields, as shown in Table 2. URL field is divided into base url and reminder, which are the host name of web server and the rest part of full URL, respectively. Then, we assume that each URL is semantically characterized by its base URL.

Table 2. Example

Timestamp	IP Address	URL(BaseURL + Reminder)
$<t_1, t_2, t_3, t_4>$	ip_1	$<b\_url_1+r_1, b\_url_1+r_2, b\_url_1+r_3, b\_url_2+r_4>$
$<t_5>$	ip_2	$<b\_url_1+r_5>$
$<t_6, t_7, t_8>$	ip_1	$<b\_url_2+r_6, b\_url_1+r_7, b\_url_3+r_7>$

For example, we are given a web log composed of eight requests ordered by timestamps from t_1 to t_8. We denote the URL set of sequential requests by $< \ldots, b\_url_i+r_j, \ldots >$ mapped to the timestamps $< \ldots, t_i, \ldots >$. These logs are partitioned with respect to an IP address ip_i. After partitioning, we compare semantic distance between base URLs in a set of requests, because we regard a semantic session as the sequence of URL having similar semantics. In other words, we investigate if an user's intention is retained or not.

3 Ontology-Oriented Heuristics for Sessionization

In order to semantically recognize what an URL is about, we try to categorize a requested URL based on ontology. Such ontologies are applied with web directories. We therefore can retrieve the tree-like conceptualized URLs, and then measure the similarity between them.

Outlier, generally, is a data object which are glossly different from or inconsistent with the remaining set of data [15]. In order to detect outliers, cluster-based and distance-based algorithms have been introduced [17], [16], [18], [19].

Meanwhile, semantic outlier of streaming web log data in this paper means a particular URL request whose semantic distance with previous sequence logs is over prefined semantic threshold. Thereby, we define these semantic measurement such as semantic mean and semantic distance. Based on semantic outliers detected by these quantified values, semantic sessionization is conducted.

3.1 URL Conceptualization Based on Web Directories

An ontology, a so-called semantic categorizer, is an explicit specification of a conceptualization [10]. It means that ontologies can play a role of enriching semantic or structural information to unlabeled data. Web directories like Yahoo[2]

[2] Yahoo. `http://www.yahoo.com`

and Cora[3] can be used to describe the content of a document in a standard and universal way as ontology [11]. Besides, web directory is organized as a topic hierarchical structure which is an efficient way to organize, view, and explore large quantities of information that would, otherwise, be cumbersome [13].

In this paper we assume that all URLs can be categorized by a well-organized web directory service. There are, however, some practical obstacles to do that, because most of web directories are forced to manage a non-generic tree structure in order to avoid a waste of memory space caused by redundant information [12]. For example, some websites for the category Computer Science:Artificial Intelligence:Constraint Satisfaction:Laboratory can also be in the category Education:Universities:Korea:Inha University:Laboratory. We briefly note that problems with categorizing an URL with web directory as an ontology are the following:

- **The multi-attributes of an URL.** An URL can be involved in more than a category. The causal relationships between categories makes their hierarchical structure more complicated. As shown in Fig. 2 (1), an URL can be included in some other categories, named as A or B.

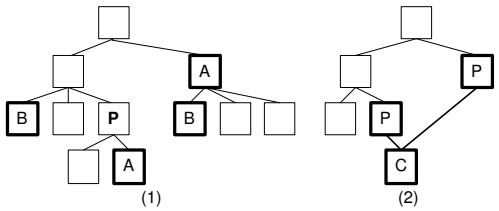


Fig. 2. (1) The multi-attribute of URLs; (2) The subordinate relationship between two categories

- **The relationship between categories.** A category can have more than a path from root node. As shown in Fig. 2 (2), the category C can be a subcategory of more than one like P. Furthermore, some categories can be semantically identical, even if they have different labels.
  - Redundancy between semantically identical categories
  - Subordination between semantically dependent categories

In order to simply handle these problems, we categorize each URL to all possible categories causally related with itself. Therefore, an URL url_i is categorized to a category set $Category(url_i)$, and the size of this category set depend on the web directory. Each element of a category set is represented as a path from the root to the corresponding category on web directory. In Table 2, let the base URLs {b_url$_1$, b_url$_2$, b_url$_3$} semantically enriched to {<a:b:d, a:b:f:k>,

[3] Cora. `http://cora.whizbang.com`

<a:c:h>, <a:b:f:j>}. The leftmost concept "a" is indicating the root of web directory and these base URLs are categorized to <d, k>, <h>, and <j>, respectively. In particular, due to multi-attribute of base URL b_url_1, $Category(\text{b\_url}_1)$ is composed of two different concepts.

3.2 Preliminary Notations and Definitions

We define semantic factors measuring the relationship between two log data. All possible categorical and ordered paths for the requested URL, above all, are obtained, after conceptualizing this URL by web directory. Firstly, the semantic distance is formulated for measuring the semantic difference between two URLs. Let an URL url_i categorized to the sets $\{path_i | path_i^m \in Category(url_i), m \in [1, \ldots, M]\}$ where M is the number of total categorical paths. As simply extending Levenshtein edit distance [21], the semantic distance $\Delta^\diamond$ between two URLs url_i and url_j is given by

$$\Delta^\diamond[url_i, url_j] = \arg \min_{m=1,n=1}^{M,N} \frac{\min\left((L_i^m - L_C^{(m,n)}), (L_j^n - L_C^{(m,n)})\right)}{\exp(L_C^{(m,n)})} \tag{1}$$

where L_i^m, L_j^n, and $L_C^{(m,n)}$ are the lengths of $path_i^m$, $path_j^n$, and common part of both of them, respectively. As marking paths representing conceptualized URLs on trees, we can easily get this common part overlapping each other. $\Delta^\diamond$ compares all combination of two sets ($|path_i| \times |path_j|$) and returns the minimum among values in the interval [0, 1], where 0 stands for complete matching. Exponent function in denominator is used in order to increase the effect of $L_C^{(m,n)}$. Second factor is to aggregate URLs during a time interval. Thereby, semantic distance matrix $D_{\Delta^\diamond}$ is given by

$$D_{\Delta^\diamond}(i,j) = \begin{bmatrix} \cdots & \cdots & \cdots \\ \cdots & \Delta^\diamond[url_{t_i}, url_{t_j}] & \cdots \\ \cdots & \cdots & \cdots \end{bmatrix} \tag{2}$$

where the predefined time interval T is the size of matrix and diagonal elements are all zero. Based on $D_{\Delta^\diamond}$, the semantic mean $\mu^\diamond$ is given by

$$\mu^\diamond(t_1, \ldots, t_T) = \frac{2\sum_{i=1}^{T}\sum_{j=i}^{T} D_{\Delta^\diamond}(i,j)}{T(T-1)} \tag{3}$$

where $D_{\Delta^\diamond}(i,j)$ is the (i,j)-th element of distance matrix. This is the mean value of upper triangular elements except diagonals. Then, with respect to the given time interval T, the semantic deviation $\sigma^\diamond$ is derived as shown by

$$\sigma^\diamond(t_1, \ldots, t_T) = \sqrt{\frac{2\sum_{i=1}^{T}\sum_{j=i}^{T} \left(D_{\Delta^\diamond}(i,j) - \mu^\diamond(t_1, \ldots, t_T)\right)^2}{T(T-1)}} \tag{4}$$

These factors are exploited to quantify the semantic distance between two random logs and statistically discriminate semantic outliers such as the most distinct or the N distinct data from the rest in the range of over pre-fixed threshold, with respect to given time interval.

3.3 Semantic Outlier Analysis for Sessionization

When we try to segment web log dataset, log entries are generally time-varying, more properly, streaming. In case of streaming dataset, not only semantic factors in a given interval but also the distribution of the semantic mean $\mu^{\diamond}$ is needed for sessionization. This will be described in the Sect. 4. We, hence, simply assume that a given dataset is time-invariant and its size is fixed in this section.

In order to analyze semantic outlier for sessionization, we regard the minimize the sum of partial semantic deviation $\mu^{\diamond}$ for each session as the most optimal partitioning of given dataset. Thereby, the principle session identifiers $PSI = \{psi_a | a \in [1, \ldots, S-1], psi_a \in [1, \ldots, T-1]\}$ is defined as the set of boundary positions, where the variables S and T are the required number of sessions and the time interval, respectively.

The semantic outlier analysis for sessionizing static logs SOA_S as objective function with respect to PSI is given by

$$SOA_S(PSI) = \sum_{i=1}^{S} \mu_i^{\diamond} \tag{5}$$

where $\mu_i^{\diamond}$ means partial semantic deviation of i^{th} segment. In order to minimize this objective function, we scan the most distinct pairs, in other words, the largest value in the semantic distance matrix $D_{\Delta^{\diamond}}$, as follows:

$$\Delta_{MAX}^{\diamond}[T_a, T_b] = \arg \max_{i=1,j=1}^{T} D_{\Delta^{\diamond}}(i, j) \tag{6}$$

where $\arg\max_{i=1}^{T}$ is the function returning the maximum values during a given time interval $[T_a, T_b]$. When we obtain $D_{\Delta^{\diamond}}(p, q)$ as the maximum semantic distance, we assume there must be at least a principle session identifier between p^{th} and q^{th} URLs. Then, the initial time interval $[T_a, T_b]$ is replaced by $[T_p, T_q]$, and the maximum semantic distance in reduced time interval is scanned, recursively. Finally, when two adjacent elements are acquired, we evaluate this candidate psi by using $SOA_S(psi)$. If this value is less than $\sigma^{\diamond}$, this cadidate psi is inserted in PSI. Otherwise, this partition by this candidate psi is cancelled. This sessionization process is top-down approaching, until the required number of sessions S is found. Furthermore, we can also be notified the oversessionization, which is a failure caused by overfitting sessionization, detected by the evaluation process $SOA_S(PSI)$.

As an example, let a URL entry composed of two sessions S_A and S_B in two cases, as shown in Fig. 3. We assume that the semantic distances between $\mathrm{A}^{(i)}$s (or $\mathrm{B}^{(i)}$s) is much less than between each other.

In the first case (Case 1), due to the maximum distance $\Delta^{\diamond}[A^{(1)}, B^{(1)}]$ in the initial time interval $[1, 6]$, time interval is reduced to $[2, 5]$, and then, $\Delta^{\diamond}[A^{(2)}, B^{(0)}]$ in updated time interval determines that psi_3 can be a candidate. Finally, the evaluation $\sigma^{\diamond}[1, 3] + \sigma^{\diamond}[4, 6] < \sigma^{\diamond}[1, 6]$ makes a candidate psi_3 inserted to PSI. This case is clear to find the cadidate psi and prove this sessionization to be validate.

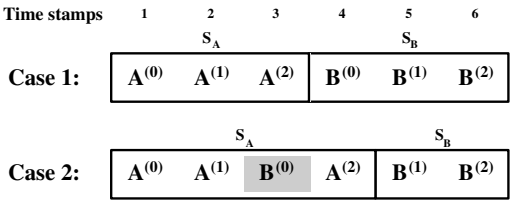


Fig. 3. Top-down approaching sessionization. The four largest semantic distances $\Delta^{\diamond}[A^{(1)}, B^{(1)}]$, $\Delta^{\diamond}[A^{(2)}, B^{(2)}]$, $\Delta^{\diamond}[A^{(2)}, B^{(0)}]$, and $\Delta^{\diamond}[A^{(2)}, B^{(1)}]$ are 0.86, 0.85, 0.81, and 0.79, respectively. We want to segement them into two sessions.

More complicatedly, in second case (Case 2), a heterogeneous request $B^{(0)}$ is located in the session S_A. The first candidate psi_3 is generated by $\Delta^{\diamond}[B^{(0)}, A^{(2)}]$ in time interval firstly refined by $\Delta^{\diamond}[A^{(1)}, B^{(1)}]$. By the evaluation $\sigma^{\diamond}[1,3] + \sigma^{\diamond}[4,6] \geq \sigma^{\diamond}[1,6]$, however, this candidate psi_3 is removed. Finally, because the second candidate psi_4 by $\Delta^{\diamond}[A^{(2)}, B^{(1)}]$ meets the evaluation $\sigma^{\diamond}[1,4]+\sigma^{\diamond}[5,6] < \sigma^{\diamond}[1,6]$, a candidate psi_4 can be into PSI.

According to the required number of sessions, this recursion process can be executed.

4 Session Identification from Streaming Web Logs

Actually, on-line web logs are continuously changing. It is impossible to consider not only the existing whole data but also streaming data. We define the time window W as the pre-determined size of considerable entry from the most recent one. Every time new URL is requested, this time window have to be shifted. In order to semantic outlier analysis of streaming logs, we focus on not only basic semantic factors but also the distribution of the semantic mean with respect to time window, $\mu^{\diamond}(W^{(T)})$.

As extending SOA_S, the objective function for analyzing semantic outlier of dynamic logs SOA_D is given by

$$SOA_D^{W^{(i)}}(PSI) = \sum_{k=1}^{S} \mu_k^{\diamond}|_{W^{(i)}} \tag{7}$$

where the $W^{(i)}$ means that the time window from i^{th} URL is applied. We want to minimize this $SOA_D(PSI)$ by finding the most proper set of principle session identifiers. The candidate psi_i is estimated by the difference between the semantic means of contiguous time windows and predefined threshold ε, as shown by

$$\left|\mu^{\diamond}(W^{(i)}) - \mu^{\diamond}(W^{(i-\tau)})\right| \geq \varepsilon \tag{8}$$

where τ is the distance between both time windows and assumed to be less than the size of time window $|W|$. Similar to the evaluation process of SOA_S,

once a candidate psi_i is obtained, we evaluate it by comparing $SOA_D^{W^{(i)}}$ and $SOA_D^{W^{(i-1)}}$. Finally, we can retrieve PSI to sessionize streaming web logs. In case of streaming logs, more particularly, a candidate psi meeting the evaluation process can be appended into unlimited size of PSI.

5 Experiments and Discussion

For experiments, we collected the sanitized access logs from sv.us.ircache.net, one of web cache servers of IRCache. These files were generated for seven days from 20 March 2003 to 26 March 2003. Raw data consist of 11 attributes, as shown in the Table 1, and about 9193000 entries. The total size of them is about 1.2 GB.

We verified sessionizing process proposed in this paper on a PC with a 1.2 GHz CPU clock rate, 256 MB main memory, and running FreeBSD 5.0. During data cleansing, logs whose URL field is ambiguous (wrong spelling or IP address) are removed, as referring to web directory.

We compared two sessionizations based on time oriented and ontology oriented heuristics, with respect to the number of segmented sessions and the reasonability of association rules extracted from them. In case of ontology-oriented sessionization, fields related with time such as "Timestamp" and "Elapsed Time" were filtered. Time-oriented heuristics simply sessionized log entries between two sequential requests whose difference of field "Timestamp" is more than 20 milliseconds with respect to the same IP address. On the other hand, for ontology-oriented heuristics, the size of time window W was predefined as 50.

Table 3. The number of sessions by time-oriented heuristics and ontology-oriented heuristics (static and dynamic logs) from logs for seven days (20-36 March 2003).

	1	2	3	4	5	6	7
Time-oriented	1563	1359	1116	877	1467	1424	1384
Ontology-oriented	907	923	692	421	807	783	844
(Static logs, SOA_S)	(58%)	(68%)	(62%)	(48%)	(55%)	(55%)	(61%)
Ontology-oriented	983	1051	939	683	1118	827	1105
(Dynamic logs, SOA_D)	(63%)	(77%)	(84%)	(78%)	(76%)	(58%)	(80%)
Common Session Boundary	47%	51%	49%	48%	57%	32%	74%

The numbers of sessions generated in both cases are shown in Table 3. Time-oriented heuristics esimate denser sessionization than two ontology-oriented approaches. It means that ontology-oriented heuristics based on SOA_S or SOA_D, generally, can make URLs requested over time gap semantically connected each other. They, SOA_S or SOA_D, decreased the number of sessions to, overall, 58.14% and 73.71%, respectively, compared to time-oriented heuristics. Even though ontology-oriented heuristics searched fewer sessions, the rate of common

session boundaries (the number of common sessions matched with time-oriented heuristics over the number of sessions of SOA_D) is average 51.1%. It shows that more than 48% of sessions not segmented by time-oriented heuristics can be detected by semantic outlier analysis. While time oriented sessionization is impossible to recognize patterns of users who is easily changing their preferences or simultaneously trying to search various kinds of information on the web, ontology-oriented method can discriminate these complicated patterns.

We also evaluated the reasonability of the rules extraced from three kinds of session sequences. According to the standard *least recently used* (LRU), we organized the expected set of URLs, which means the set of objects that cache server has to prefetch. The size of this set is constantly 100.

Table 4. Evaluation of the reasonability of the extracted ruleset (hit ratio (%))

	1	2	3	4	5	6	7
Time-oriented	0.06	0.32	0.46	0.41	0.51	**0.52**	0.49
Static logs, SOA_S	0.05	0.45	0.66	0.72	**0.76**	0.74	0.75
Dynamic logs, SOA_D	0.05	0.46	0.52	0.67	0.70	**0.75**	0.72

As shown in Table 4, we measured the two hit ratios by both of their sessionizations for seven days. The maximum hit ratios in three sequences were obtained 0.52, 0.76, and 0.75, respectively. Ontology-oriented sessionization SOA_S acquired about 24.5% improvement of prefetching performance, compared with time-oriented. Moreover, we want to note that the difference between SOA_S and SOA_D. For the first three days, the hit ratio of SOA_S was higher than that of SOA_D by over 5%. Because of streaming data, SOA_D showed the difficulty in initializing the ruleset. After initialization step, however, the performances of SOA_S and SOA_D were converged into a same level.

6 Conclusions and Future Work

In order to mine useful and significant association rules from web logs, many kinds of well-known association discoverying methods have been developed. Due to the domain specific properties of web logs, sessionization process of log entries is the most important in a whole step. We have proposed ontology-oriented heuristics for sessionizing web logs. In order to provide each requested URL with the corresponding semantics, web directory service as ontology have been applied to categorize this URL. Especially, we mentioned three practical problems for using real non-generic tree structured web directories like Yahoo. After conceptualizing URLs, we measured the semantic distance matrix indicating the relationships between URLs within the predefined time interval. Additionally, factors like semantic mean and semantic deviation were formulated for easier computation.

We considered two kinds of web logs which are stationary and streaming. Therefore, two semantic outlier analysis approaches SOA_S and SOA_D were introduced based on semantic factors. Through the evaluation process, the detected cadidate semantic outliers were tested whether their sessionization is reasonable or not. According to results of our expements, investigating semantic relationships between web logs is very important to sessionize them. Classifying semantic sessions, 48% of total sessions, brought about 25% higher prefetching performance, campared with time-oriented sessionization. Complex web usage patterns seemed to be meaninglessly mixed along with "time" can be analyzed by ontology.

In future work, we have to consider bottom-up approaching sessionization to cluster sessions divided by top-down approaching. As exploiting semantic sessionization proposed in this paper to the web proxy server, more practically, we will study association rule mining on various web caching architecture [14], in order to improve the predicability of content prefection.

Acknowledgement. This work was supported by INHA UNIVERSITY Research Grant. (INHA-30305)

References

1. IRCache Users Guide. http://www.ircache.net/
2. Cooley, R., Srivastava, J., Mobasher, B.: Web Mining: Information and Pattern Discovery on the World Wide Web. Proc. of the 9th IEEE Int. Conf. on Tools with Artificial Intelligence (1997)
3. Mobasher, B., Cooley, R., Srivastava, J.: Automatic personalization based on Web usage mining. Communications of the ACM **43**(8) (2000)
4. Berendt, B., Mobasher, B., Nakagawa, M., Spiliopoulou, M.: The Impact of Site Structure and User Environment on Session Reconstruction in Web Usage Analysis. Proc. of the 4th WebKDD Workshop at the ACM-SIGKDD Conf. on Knowledge Discovery in Databases (2002)
5. Chen, Z., Tao, L., Wang, J., Wenyin, L., Ma, W.-Y.: A Unified Framework for Web Link Analysis. Proc. of the 3rd Int. Conf. on Web Information Systems Engineering (2002) 63–72
6. Batista, P., Silva, M.J.: Web Access Mining from an On-line Newspaper Logs. Proc. 12th Int. Meeting of the Euro Working Group on Decision Support Systems (2001)
7. Bonchi, F., Giannotti, F., Gozzi, C., Manco, G., Nanni, M., Pedreschi, D., Renso, C., Ruggieri, S.: Web log data warehousing and mining for intelligent web caching. Data and Knowledge Engineering **39**(2) (2001) 165–189
8. Cooley, R., Mobasher, B., Srivastava, J.: Data Preparation for Mining World Wide Web Browsing Patterns. Knowledge and Information Systems **1**(1) (1999) 5–32
9. Dai, H., Mobasher, B.: Using ontologies to discover domain-level web usage profiles. Proc. of the 2nd Semantic Web Mining Workshop at the PKDD 2002 (2002)
10. Gruber, T.: What is an Ontology? http://www-ksl.stanford.edu/kst/what-is-an-ontology.html

11. Labrou, Y., Finin, T.: Yahoo! as an Ontology: Using Yahoo! Categories to Describe Documents. Proc. of the 8th Int. Conf. on Information Knowledge Management (1999) 180–187
12. Jung, J.J., Yoon, J.-S., Jo, G.-S.: Collaborative Information Filtering by Using Categorized Bookmarks on the Web. Proc. of the 14th Int. Conf. on Applications of Prolog (2001) 343–357
13. McCallum, A., Nigam, K., Rennie, J., Seymore, K.: Building Domain-Specific Search Engines with Machine Learning Techniques. AAAI Spring Symposium (1999)
14. Aggarwal, C., Wolf, J.L., Yu, P.S.: Caching on the World Wide Web. IEEE Tran. on Knowldge and Data Engineering **11**(1) (1999) 94–107
15. Han, J., Kamber, M.: Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers (2001)
16. Arning, A., Agrawal, R., Raghavan, P.: A Linear Model for Deviation Detection in Large Databases. Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining, (1996) 164–169
17. Ramaswamy, S., Rastogi, R., Shim, K.: Efficient Algorithms for Mining Outliers from Large Data Sets. Proc. of the ACM SIGMOD Conf. on Management of Data (2000) 427–438
18. Menasalvas, E., Millan, S., Pena, J.M., Hadjimichael, M., Marban, O.: Subsessions: a granular approach to click path analysis. Proc. of the IEEE Int. Conf. on Fuzzy Systems (2002) 878–883
19. Pei, J., Han, J., Mortazavi-Asl, B., Zhu, H.: Mining Access Patterns Efficiently from Web Logs. Proc. of Pacific-Asia Conf. on Knowledge Discovery and Data Mining (PAKDD'00) (2000)
20. Berendt, B., Spiliopoulou, M.: Analysing navigation behaviour in web sites integrating multiple information systems. The VLDB Journal **9**(1) (2000) 56–75
21. Levenshtein, I.V.: Binary Codes capable of correcting deletions, insertions, and reversals. Cybernetics and Control Theory, **10**(8) (1966) 707–710

Construction of Web Community Directories using Document Clustering and Web Usage Mining

Dimitrios Pierrakos[1], Georgios Paliouras[1], Christos Papatheodorou[2], Vangelis Karkaletsis[1], Marios Dikaiakos[3]

[1] Institute of Informatics and Telecommunications, NCSR "Demokritos", 15310 Ag. Paraskevi, Greece
{dpie, paliourg, vangelis}@iit.demokritos.gr
[2] Department of Archive & Library Sciences, Ionian University 49100, Corfu, Greece
papatheodor@ionio.gr
[3] Department of Computer Science, University of Cyprus CY1678, Nicosia, Cyprus
mdd@ucy.ac.cy

Abstract. This paper presents the concept of Web Community Directories, as a means of personalizing services on the Web, together with a novel methodology for the construction of these directories by document clustering and usage mining methods. The community models are extracted with the use of the Community Directory Miner, a simple cluster mining algorithm which has been extended to ascend a concept hierarchy, and specialize it to the needs of user communities. The initial concept hierarchy is generated by a content-based document clustering method. Communities are constructed on the basis of usage data collected by the proxy servers of an Internet Service Provider. These data present a number of peculiarities such as their large volume and semantic diversity. Initial results presented in the paper illustrate the use of the methodology and provide an indication of the behavior of the new mining method.

1 Introduction

The hypergraphical architecture of the Web has been used to support claims that the Web will make Internet-based services really user-friendly. However, at its current state, the Web has not achieved its goal of providing easy access to online information. Being an almost unstructured and heterogeneous environment it creates an information overload and places obstacles in the way users access the required information.

One approach towards the alleviation of this problem is the organization of Web content into thematic hierarchies, also known as *Web directories*. A Web directory, such as Yahoo! [19] or the Open Directory Project (ODP) [14], allows Web users to locate information that relates to their interests, through a hierarchy navigation process. This approach suffers though from a number of problems. The manual creation and maintenance of the Web directories leads to limited coverage of the topics that are contained in those directories, since there are millions of Web pages and the rate of

expansion is very high. In addition, the size and complexity of the directories is canceling out any gains that were expected with respect to the information overload problem, i.e., it is often difficult for a particular user to navigate to interesting information.

An alternative solution is the personalization of the services on the Web. *Web Personalization* [10] focuses on the adaptability of Web-based information systems to the needs and interests of individuals or groups of users and aims to make the Web a friendlier environment. Typically, a personalized Web site recognizes its users, collects information about their preferences and adapts its services, in order to match the users' needs. A major obstacle towards realizing Web personalization is the acquisition of accurate and operational models for the users. Reliance to manual creation of these models, either by the users or by domain experts, is inadequate for various reasons, among which the annoyance of the users and the difficulty of verifying and maintaining the resulting models. An alternative approach is that of *Web Usage Mining* [18], which uses data mining methods to create models, based on the analysis of usage data, i.e., records of how a service on the Web is used. Web usage mining provides a methodology for the collection and preprocessing of usage data, and the construction of models representing the behavior and the interests of users [16].

In this paper, we propose a solution to the problem of information overload, by combining the strengths of Web Directories and Web Personalization, in order to address some of the above-mentioned issues. In particular we focus on the construction of usable Web directories that correspond to the interests of groups of users, known as *user communities*. The construction of user community models with the aid of Web Usage Mining has so far only been studied in the context of specific Web sites [15]. This approach is extended here to a much larger portion of the Web, through the analysis of usage data collected by the proxy servers of an Internet Service Provider (ISP). The final goal is the construction of community-specific Web Directories. Web Community Directories can be employed by various services on the Web, such as Web portals, in order to offer their subscribers a more personalized view of the Web. The members of a community can use the community directory as a starting point for navigating the Web, based on the topics that they are interested in, without the requirement of accessing vast Web directories. In this manner, the information overload is reduced, while at the same time the service offers added value to its customers.

The construction of community directories with usage mining raises a number of interesting research issues, which are addressed in this paper. The first challenge is the analysis of large datasets in order to identify community behavior. In addition to the heavy traffic expected at a central node, such as an ISP, a peculiarity of the data is that they do not correspond to hits within the boundaries of a site, but record outgoing traffic to the whole of the Web. This fact leads to the increased dimensionality and the semantic incoherence of the data, i.e., the Web pages that have been accessed. In order to address these issues we create a thematic hierarchy of the Web pages by examining their content, and assign the Web pages to the categories of this hierarchy. An agglomerative clustering approach is used to construct the hierarchy with nodes representing content categories. A community construction method then exploits the constructed hierarchy and specializes it to the interests of particular communities. The basic data mining algorithm that has been developed for that purpose, the *Community Directory Miner* (CDM), is an extension of the *cluster mining* algorithm, which has

been used for the construction of site-specific communities in previous work [15]. The new method proposed here is able to ascend an existing directory in order to arrive at a suitable level of semantic characterization of the interests of a particular community.

The rest of this paper is organized as follows. Section 2 presents existing approaches to Web personalization with usage mining methods that are related to the work presented here. Section 3 presents in detail our methodology for the construction of Web community directories. Section 4 provides results of the application of the methodology to the usage data of an ISP. Finally section 5 summarizes the most interesting conclusions of this work and presents promising paths for future research.

2 Related Work

In recent years, the exploitation of usage mining methods for Web personalization has attracted considerable attention and a number of systems use information from Web server log files to construct user models that represent the behavior of the users. Their differences are in the method that they employ for the construction of user models, as well as in the way that this knowledge, i.e., the models, is exploited. Clustering methods, e.g. [7], [11] and [20], classification methods, e.g. [13], and sequential pattern discovery, e.g. [17], have been employed to create user models. These models are subsequently used to customize the Web site and recommend links to follow. Usage data has also been combined with the content of Web pages in [12]. In this approach content profiles are created using clustering techniques. Content profiles represent the users' interests for accessed pages with similar content and are combined with usage profiles to support the recommendation process. A similar approach is presented in [6]. Content and usage data are aggregated and clustering methods are employed for the creation of richer user profiles. In [4], Web content data are clustered for the categorization of the Web pages that are accessed by users. These categories are subsequently used to classify Web usage data.

Personalized Web directories, on the other hand, are mainly associated with services such as Yahoo! [19] and Excite [5], which support manual personalization by the user. An initial approach to automate this process, with the aid of usage mining methods, is the Montage system [1]. This system is used to create personalized portals, consisting primarily of links to the Web pages that a particular user has visited, organized into thematic categories according to the ODP directory. For the construction of the user model a number of heuristic metrics are used, such as the interest in a page or a topic, the probability of revisiting a page, etc. An alternative approach is the construction of a directory of useful links (bookmarks) for an individual user, as adopted by the PowerBookmarks system [8]. The system collects "bookmark" information for a particular user, such as frequently visited pages, query results from a search engine, etc. Text classification techniques are used for the assignment of labels to Web pages. An important issue regarding these methods is the scalability of the classification methods that they use. These methods may be suitable for constructing models of what a single user usually views, but their extendibility to aggregate user models is questionable. Furthermore, the requirement for a small set of predefined classes complicates the construction of rich hierarchical models.

In contrast to existing work, this paper proposes a novel methodology for the construction of Web directories according to the preferences of user communities, by combining document clustering and usage mining techniques. A hierarchical clustering method is employed for document clustering using the content of the Web pages. Subsequently, the hierarchy of document categories is exploited by the Web usage mining process and the complete paths of this hierarchy are used for the construction of Web community directories. This approach differs from the related work mentioned above, where the content of the Web pages is clustered in order to either enhance the user profiles or to assign Web usage data to content categories.

The community models are aggregate user models, constructed with the use of a simple cluster mining method, which has been extended to ascend a concept hierarchy, such as a Web directory, and specialize it to the preferences of the community. The construction of the communities is based on usage data collected by the proxy servers of an Internet Service Provider (ISP), which is also a task that has not been addressed adequately in the literature. This type of data has a number of peculiarities, such as its large volume and its semantic diversity, as it records the navigational behavior of the users throughout the Web, rather than within a particular Web site. The methodology presented here addresses these issues and proposes a new way of exploiting the extracted knowledge. Instead of link recommendation or site customization, it focuses on the construction of Web community directories, as a new way of personalizing services on the Web.

3 Constructing Web community Directories

The construction of Web community directories is seen here as the end result of a usage mining process on data collected at the proxy servers of a central service on the Web. This process consists of the following steps:

- *Data Collection and Preprocessing*, comprising the collection and cleaning of the data, their characterization using the content of the Web pages, and the identification of user sessions. Note that this step involves a separate data mining process for the discovery of content categories and the characterization of the pages.
- *Pattern Discovery*, comprising the extraction of user communities from the data with a suitably extended cluster mining technique, which is able to ascend a thematic hierarchy, in order to discover interesting patterns.
- *Knowledge Post-Processing*, comprising the translation of community models into Web community directories and their evaluation.

An architectural overview of the discovery process is given in Figure 1, and described in the following sections.

3.1 Data Collection and Preprocessing

The usage data that form the basis for the construction of the communities are collected in the access log files of proxy servers, e.g. ISP cache proxy servers. These data record the navigation of the subscribers through the Web. No record of the user's identification is being used, in order to avoid privacy violations. However, the data

collected in the logs are usually diverse and voluminous. The outgoing traffic is much higher than the usual incoming traffic of a Web site and the visited pages less coherent semantically. The task of data preprocessing is to assemble these data into a consistent, integrated and comprehensive view, in order to be used for pattern discovery.

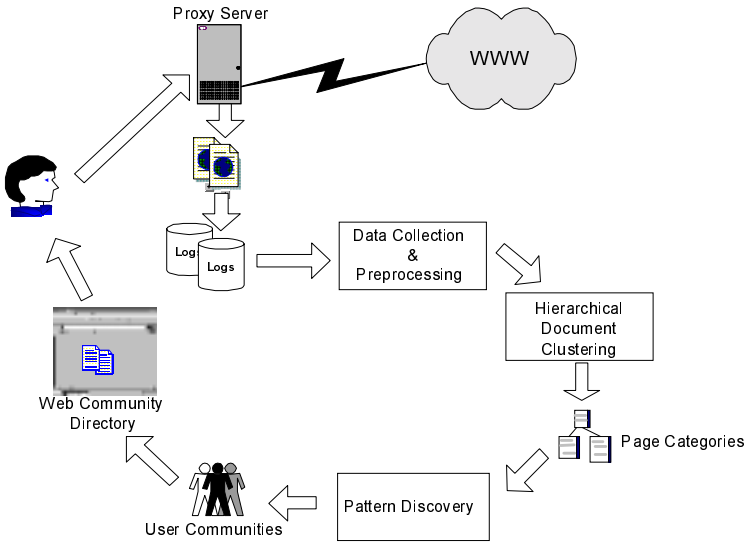


Fig. 1. The process of constructing Web Community Directories.

The first stage of data preprocessing involves data cleaning. The aim is to remove as much noise from the data as possible, in order to keep only the Web pages that are directly related to the user behavior. This involves the filtering of the log files to remove data that are downloaded without a user explicitly requesting them, such as multimedia content, advertisements, Web counters, etc. Records with HTTP error codes that correspond to bad requests, or unauthorized accesses are also removed.

The second stage of data preprocessing involves the thematic categorization of Web pages, thus reducing the dimensionality and the semantic diversity of data. Typically, Web page categorization approaches, e.g. [3] and [9], use text classification methods to construct models for a small number of known thematic categories of a Web directory, such as that of Yahoo!. These models are then used to assign each visited page to a category. The limitation of this approach with respect to the methodology proposed here, is that it is based on a dataset for training the classifiers, which is usually limited in scope, i.e., covers only part of the directory. Furthermore, a manually-constructed Web directory is required, suffering from low coverage of the Web.

In contrast to this approach, we build a taxonomy of Web pages included in the log files. This is realized by a document clustering approach, which is based on terms that are frequently encountered in the Web pages. Each Web page is represented by a

binary feature vector, where each feature encodes the presence of a particular term in the document. A hierarchical agglomerative approach [21] is employed for document clustering. The nodes of the resulting hierarchy represent clusters of Web pages that form thematic categories. By exploiting this taxonomy, a mapping can be obtained between the Web pages and the categories that each page is assigned to. Moreover, the most important terms for each category can be extracted, and be used for descriptive labeling of the category. For the sake of brevity we choose to label each category using a numeric coding scheme, representing the path from the root to the category node, e.g. "1.4.8.19" where "1" corresponds to the root of the tree.

This document clustering approach has the following advantages: first a hierarchical classification of Web documents is constructed without any human expert intervention or other external knowledge; second the dimensionality of the space is significantly reduced since we are now examining the page categories instead of the pages themselves; and third the thematic categorization is directly related to the preferences and interests of the users, i.e. the pages they have chosen to visit.

The third stage of preprocessing involves the extraction of access sessions. An access session is a sequence of log entries, i.e., accesses to Web pages by the same IP address, where the time interval between two subsequent entries does not exceed a certain time interval. In our approach, pages are mapped onto thematic categories and therefore an access session is translated into a sequence of categories. Access sessions are the main input to the pattern discovery phase, and are extracted as follows:

1. Grouping the logs by date and IP address.
2. Selecting a time-frame within which two records from the same IP address can be considered to belong in the same access session.
3. Grouping the Web pages (thematic categories) accessed by the same IP address within the selected time-frame to form a session.

Finally, access sessions are translated into binary feature vectors. Each feature in the vector represents the presence of a category in that session.

3.2 Extraction of Web Communities

Once the data have been translated into feature vectors, they are used to discover patterns of interest, in the form of community models. This is done by the *Community Directory Miner* (CDM), an enhanced version of the cluster mining algorithm. This approach is based on the work presented in [15] for site-specific communities.

Cluster mining discovers patterns of common behavior by looking for all maximal fully-connected subgraphs (cliques) of a graph that represents the users' characteristic features, i.e., thematic categories in our case. The method starts by constructing a weighted graph $G(A,E,W_A,W_E)$. The set of vertices A corresponds to the descriptive features used in the input data. The set of edges E corresponds to feature co-occurrence as observed in the data. For instance, if the user visits pages belonging to categories "1.3.5" and "1.7.8" an edge is created between the relevant vertices. The weights on the vertices W_A and the edges W_E are computed as the feature occurrence and co-occurrence frequencies respectively.

Figure 2 shows an example of such a graph. The connectivity of the graph is usually very high. For this reason we make use of a *connectivity threshold* aiming to re-

duce the edges of the graph. This threshold is related to the frequency of the thematic categories in the data. In our example in Figure 2, if the threshold is 0.07 the edge ("1.3.5", "1.3.6") is dropped. Once, the connectivity of the graph has been reduced, all maximal cliques of the graph are generated, each one corresponding to a community model. One important advantage of this approach is that each user may be assigned to many communities, unlike most user clustering methods.

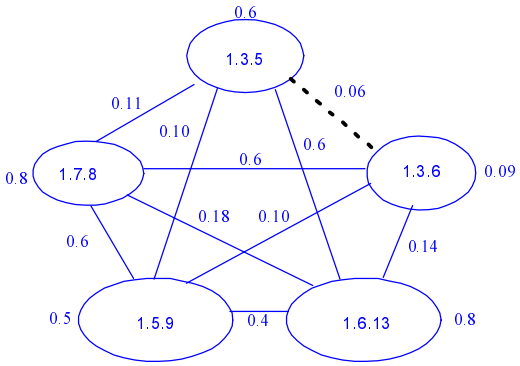


Fig. 2. An example of a graph for cluster mining.

CDM enhances cluster mining so as to be able to ascend a hierarchy of topic categories. This is achieved by updating the weights of the vertices and the nodes in the graph. Initially, each category is mapped onto a set of categories, corresponding to its parent and grandparents in the thematic hierarchy. Thus, the category "1.6.12.122.258" is also mapped onto the following categories: "1", "1.6", "1.6.12", "1.6.12.122". The frequency of each of these categories is increased by the frequency of the initial child category. Thus, the frequency of each category corresponds to its own original frequency, plus the frequency of its children. The underlying assumption for the update of the weights is that if a certain category exists in the data, then its parent categories should also be examined for the construction of the community model. In this manner, even if a category (or a pair of categories) have a low occurrence (co-occurrence) frequency, their parents may have a sufficiently high frequency to be included in a community model. This enhancement allows the algorithm to start from a particular category and ascend the topic hierarchy accordingly. The result is the construction of a topic tree, even if only a few nodes of the tree exist in the usage data.

The CDM algorithm can be summarized in the following steps:

Step 1: *Compute frequencies of categories that correspond to the weights of the vertices*. More formally, if a_{ij} is the value of a feature i in the binary feature vector j, and there are N vectors, the weight of w_i for that vertice is calculated as follows:

$$w_i = \frac{\sum_{j=1}^{N} a_{ij}}{N}. \qquad (1)$$

Step 2: *Compute co-occurrence frequencies between categories that correspond to the weights of the edges.* If a_{ik}^{j} is a binary indicator of whether features i and k co-occur in vector j, then the weight of the edge w_{ik} is calculated as follows:

$$w_{ik} = \frac{\sum_{j=1}^{N} a_{ik}^{j}}{N}. \qquad (2)$$

Step 3: *Update the weights of categories, i.e. vertices, by adding the frequencies of their children.* More formally, if w_p is the weight of a parent vertex p and w_i is the weight of a child vertex i, the final weight w'_p of the parent is computed as follows:

$$w'_p = w_p + \sum_{i} w_i \qquad (3)$$

This calculation is repeated recursively ascending the hierarchy of the Web directory. Similarly, the edge weights are updated, as all the parents and grandparents of the categories that co-occur in a session, are also assumed to co-occur.

Step 4: *Find all the maximal cliques in the graph of categories* [2].

3.3 Post-Processing and Model Evaluation

The discovered patterns are topic trees, representing the community models, i.e., behavioral patterns that occur frequently in the data. These models are directly usable as Web community directories, and can be delivered by various means to the users of a community. A pictorial view of such a directory is shown in Figure 3, where the community directory is "superimposed" onto the hierarchy of categories. Grey boxes represent the categories that belong to a particular community, while white boxes represent the rest of the categories in the Web directory. Each category has been labeled using the most frequent terms of the Web pages that belong to this category. The categories "12.2", "18.79" and "18.85" appear in the community model, due to the frequency of their children. Furthermore, some of their children, e.g. "18.79.5" and "18.79.6" (the spotted boxes) may also not be sufficiently frequent to appear in the model. Nevertheless, they force their parent category, i.e., "18.79" into the model.

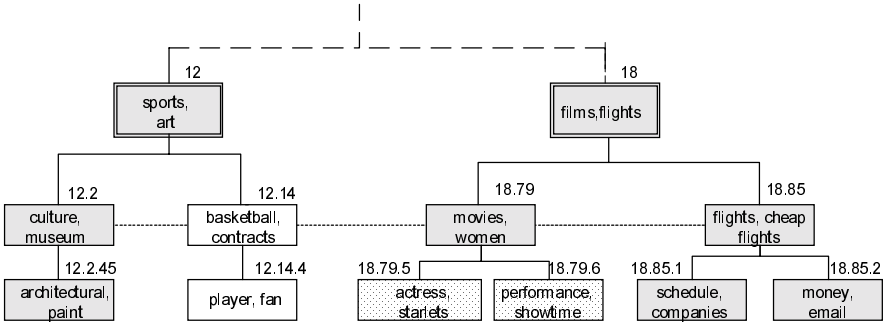


Fig. 3. An example of a Web community directory.

Having generated the community models, we need to decide on their desired properties, in order to evaluate them. For this purpose, we use ideas from existing work on community modeling and in particular the measure of *distinctiveness* [15]. When there are only small differences between the models, accounting for variants of the same community, the segmentation of users into communities is not interesting. Thus, we are interested in community models that are as distinct from each other as possible. We measure the distinctiveness of a set of models M by the ratio between the number of distinct categories that are covered and the number of models in M. Thus, if J the number of models in M, A_j the categories used in the j-th model, and A' the different categories appearing at least in one model, distinctiveness is given by equation 4.

$$Distinctiveness(M) = \frac{|A'|}{\sum_{j} |A_j|}. \qquad (4)$$

The optimization of distinctiveness by a set of community models indicates the presence of useful knowledge in the set. Additionally, the number of distinct categories A' that are used in a set of community models is also of interest as it shows the extent to which there is a focus on a subset of categories by the users. These two measures are used in the experimental results presented in the following section.

4. Experimental Results

The methodology introduced in this paper for the construction of Web community directories has been tested in the context of a research project, which focuses on the analysis of usage data from the proxy server logs of an Internet Service Provider. We analyzed log files consisting of 781,069 records, and the results are presented here.

In the stage of pre-processing, data cleaning has been performed and the remaining data has been characterized using the hierarchical agglomerative clustering mentioned in section 3.1. The process resulted in the creation of 998 distinct categories. Based on these characterized data, we constructed 2,253 user sessions, using a time-interval of 60 minutes as a threshold on the "silence" period between two consecutive requests from the same IP. After mapping the Web pages of the sessions to the categories of the hierarchy, we translated the sessions into binary vectors and analyzed them by the CDM algorithm, in order to identify community models, in the form of topic trees. The resulting models were evaluated using the two measures that were mentioned in section 3.3, i.e. the distinctiveness and the number of distinct categories, while varying the connectivity threshold. Figures 4 and 5 present the results of this process.

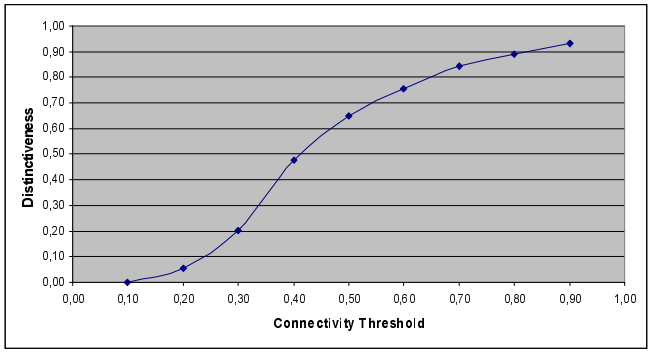


Fig. 4. Distinctiveness as a function of the connectivity threshold.

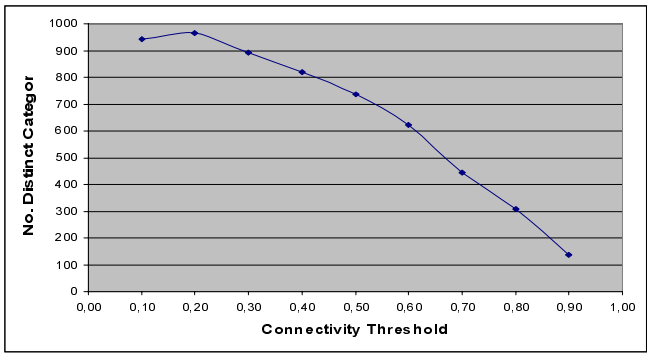


Fig. 5. Number of distinct categories as a function of the connectivity threshold.

Figure 4 shows how the distinctiveness of the resulting community models increases as the connectivity threshold increases, i.e., as the requirement on the frequency of occurrence/co-occurrence becomes "stricter". The rate of increase is higher for smaller values of the threshold and starts to dampen down for values above 0.7. This effect is justified by the decrease in the number of distinct categories, as shown in Figure 5. Nevertheless, more than half of the categories have frequency of occurrence greater than 600 (threshold 0.6), while at that threshold value the level of distinctiveness exceeds 0.7, i.e. 70% of the categories that appear in the model are distinct. These figures provide an indication of the behavior and the effectiveness of the community modeling algorithm. At the same time, they assist in selecting an appropriate value for the connectivity threshold and a corresponding set of community models. The results are encouraging for the exploitation of the proposed methodology.

5. Conclusions and Future Work

This paper has presented a novel methodology for the personalization of Web Directories with the aid of document clustering and Web usage mining. The concept of a Web community directory has been described, corresponding to a usable directory of the Web, customized to the needs and preferences of user communities. User community models take the form of thematic hierarchies and are constructed by a cluster mining algorithm, which has been extended to take advantage of an existing directory, and ascend its hierarchical structure. The initial directory is generated by a document clustering algorithm, based on the content of the pages appearing in an access log.

We have tested this methodology by applying it on access logs collected at the proxy servers of an ISP and have provided initial results, indicative of the behavior of the mining algorithm. Proxy server logs have introduced a number of interesting challenges, such as their size and their semantic diversity. The proposed methodology handles these problems by reducing the dimensionality of the problem, through the categorization of individual Web pages into the categories of a Web directory, as constructed by document clustering. In this manner, the corresponding community models take the form of thematic hierarchies.

The combination of two different approaches to the problem of information overload on the Web, i.e. thematic hierarchies and personalization, as proposed in this paper, introduces a promising research direction, where many new issues arise. Various components of the methodology could be replaced by a number of alternatives.

For instance, other mining methods could be adapted to the task of discovering community directories and compared to the algorithm presented here. Similarly, different methods of constructing the initial thematic hierarchy could be examined. Finally, additional evaluation is required, in order to test the robustness of the mining algorithm to a changing environment and the usability of the resulting community directories.

Acknowledgements

This research has been partially funded by the Greece-Cyprus Research Cooperation project "Web-C-Mine: Data Mining from Web Cache and Proxy Log Files".

REFERENCES

1. Anderson, C. R. and Eric Horvitz.: Web Montage: A Dynamic Personalized Start Page. In Proceedings of the 11th WWW Conference 2002, (2002)
2. Bron, C., Kerbosch, J.: Algorithm 457---finding all cliques of an undirected graph. Communications of the ACM, 16, 9, 575-577, (1973)
3. Chen, H., Dumais, S. T.: Bringing order to the web: automatically categorizing search results. In Proceedings of CHI'00, Human Factors in Computing Systems, 145-152, (2000)
4. Cooley, R.: Web Usage Mining: Discovery and Application of Interesting Patterns from Web Data. Ph.D. Thesis, University of Minnesota, (2000).
5. Excite, http://www.excite.com
6. Heer, J., Chi, Ed H.: Identification of Web User Traffic Composition using Multi-Modal Clustering and Information Scent. In Proceedings of the Workshop on Web Mining, SIAM Conference on Data Mining, 51-58. (2001)
7. Kamdar, T., Joshi, A.: On Creating Adaptive Web Sites using WebLog Mining. TR-CS-00-05. Department of Computer Science and Electrical Engineering University of Maryland, Baltimore County, (2000)
8. Li W-S., Vu, Q., Chang, E., Agrawal, D., Hara, Y., Takano, H.: PowerBookmarks: A System for Personalizable Web Information Organization, Sharing, and Management. In Proceedings of the 8th WWW Conference, (1999)
9. Mladenic, D.: Turning Yahoo into an Automatic Web-Page Classifier. In Proceedings of the 13th European Conference on Artificial Intelligence, 473-474, (1998)
10. Mobasher, B., Cooley, R., Srivastava, J.: Automatic personalization based on Web usage mining. TR-99010, Department of Computer Science. DePaul University, (1999)
11. Mobasher, B., Cooley, R., Srivastava, J.: Creating Adaptive Web Sites Through Usage-Based Clustering of URLs. In Proceedings of the 1999 IEEE Knowledge and Data Engineering Exchange Workshop, (1999)
12. Mobasher, B., H. Dai, T. Luo, Y. Sung, J. Zhu.: Integrating Web Usage and Content Mining for More Effective Personalization. In Proceedings of the International Conference on E-Commerce and Web Technologies. Greenwich, UK, 165-176, (2000)
13. Ngu, D. S. W., Wu, X.: SiteHelper: A Localized Agent that Helps Incremental Exploration of the World Wide Web. Computer Networks and ISDN Systems: The International Journal of Computer and Telecommunications Networking, 29, 8, 249-1255, (1997)
14. Open Directory Project (ODP). http://dmoz.org
15. Paliouras, G., Papatheodorou, C., Karkaletsis, V, Spyropoulos, C.D.: Discovering User Communities on the Internet using Unsupervised Machine Learning Techniques., Interacting with Computers Journal, 14,6, 761-791, (2002)

16. Pierrakos, D., Paliouras, G., Papatheodorou, C., Spyropoulos, C.D.: Web Usage Mining as a Tool for Personalization: a survey , User Modeling and User-Adapted Interaction, to appear
17. Spiliopoulou, N., Faulstich, L. C.: WUM: A Web Utilization Miner. In International Workshop on the Web and Databases. Valencia, Spain, (1998)
18. Srivastava, J., Cooley, R., Deshpande, M., Tan, P. T.: Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data. In SIGKDD Explorations, 1, 2, (2000)
19. Yahoo! http://www.yahoo.com
20. Yan, T. W., Jacobsen, M., Garcia-Molina, H., Dayal, U.: From User Access Patterns to Dynamic Hypertext Linking. In Proceedings of the 5th WWW Conference, Paris, France, (1996)
21. Zhao, Y., Karypis, G.: Evaluation of hierarchical clustering algorithms for document datasets. In CIKM, (2002)

Language Models for Intelligent Search Using Multinomial PCA

Wray Buntine, Petri Myllymaki and Sami Perttu

Complex Systems Computation Group (CoSCo), Helsinki Institute for Information Technology (HIIT)
University of Helsinki & Helsinki University of Technology
P.O. Box 9800, FIN-02015 HUT, Finland.
{Firstname}.{Lastname}@hiit.fi

Abstract. Once one gets beyond the simple keyword searches of internet search engines, Information Retrieval offers more extensive interfaces for searching for relevant documents. Language models for search have been developed recently. In this paper, we show how to modify these models to arrive at something which automatically incorporates learnt knowledge about topics and words in a collection. This requires, however, richer document models than are usually available. Multinomial PCA is the multinomial analogue to Gaussian PCA and is also a natural extension to clustering models but particularly suited to the bag of words model for text. Multinomial PCA turns out to be suitable for the probabilistic task of language modelling. In this paper we present the theory and demonstrate its use on a corpus of 800,000 news articles. While we do not claim this is an ideal approach (for instance, it does not use linguistic knowledge), and our demonstration is not on true HTML data (which is complicated by its poor spelling and large use of named entities), the results are very good it shows that more meaningful search is now realizable. The browser being developed for this research is available now under Open Source.

1 Introduction

Over the last decade there has been enormous growth in statistical natural language processing (e.g., [13]), and since the advent of the Web an explosion of interest in text/HTML/XML based information retrieval and information extraction. *Information retrieval* is generally viewed as the task of "find documents or parts of documents that interest me," where the user supplies a query of some kind to express their interest area. In contrast, *Information extraction* is the more extensive mining for specific things such as addresses, bibliographies or corporate records. *Question answering*, a third area is different again and is to answer a specific question using document contents.

While probabilistic methods caste in a traditional linguistic framework is arguably the dominant paradigm for natural language processing (NLP), and for information extraction, the same cannot yet be said for information retrieval. The

emerging dominant paradigm for question answering is to couple an information retrieval system using NLP essentially for data cleaning with a special purpose theorem prover using a thesaurus as a poor-man's general purpose ontology [15]. A number of frameworks have been proposed for information retrieval, and the most recent probabilistic models (e.g., [11]) are models that include some kind of smoothing or mixing that lies outside the normal rules of probability theory.

Our methodology is to first develop a probabilistic scheme for information retrieval, and then gradually augment it with NLP methods. We report on the first steps here. The probabilistic method provides the mathematical framework for devising a matching algorithm between documents. It should be viewed as complementary to techniques such as NLP or ontology based approaches. The probabilistic method provides the data fusion calculus so that other techniques can be combined with general matching principles.

In this paper we present a model for retrieval based an a probability inference task using a Multinomial PCA model for the document collection [3]. The basic idea is simple: what is the probability that the text of the query would match the text of the document or could match reasonable variations of the same document? That is, we suppose that each document in the collection has some hidden process responsible for generating it and see how well the new query (a few words, a paragraph, or new document) could have also been generated by this same process. Now we do not claim to have a sophisticated model for the generating documents, the "process". The bag of words model is trivial at best, and for instance destroys locality information used in most search engines, and destroys the linguistic clues necessary to disambiguate words. But we do claim that our demonstration of richer query modelling is well on the way to intelligent search (something that cannot be said for keyword search), something badly needed for enterprise information systems and intranets.

2 Retrieval Methods

Retrieval operates through keyword, boolean query or weighted query searches, using an inverted index, or full-text matching which generally requires a full pass of the document database. Hybrids can be done, and are used by some meta-search engines: keyword search produces candidates and full-text matching scores them.

The simple approach of keyword search coupled with all sorts of ranking to boost title words and neighbouring words is remarkably robust. For instance, when coupled with the strategy of only showing the top few matches from a single site, the problem of boiler-plates or templates (for instance, where a menu panel appears duplicated on all pages) is effectively handled. Extensions to the basic keyword approach include query expansion and document expansion. Query expansion means augmenting a query with additional words, or modifying the format. For instance, one might expand "cell phone" to "(cell OR mobile) AND phone". Document expansion does the expansion in the index, and thus has a cost in terms of space. The astute reader might notice that these are not models

in the traditional sense, but implementation schemes. Carefully crafted studies show the approach works well, but the difficult questions are often unanswered. Which expansions should be done, when, and how? A good model should provide the strategy.

Full text matching, on the other hand, requires some kind of distance measure between documents. The tried and true measure in use here is to convert the document to a set of word scores–TF/IDF score or a square-root of frequency–and apply a cosine metric [13]. More generally, measures for text is a field that is still undergoing a discovery phase, with extensive empirical work trying out different methods [5]. These methods sometimes have intuitive probability justifications (e.g., [6]). Considerably more sophisticated probability scores have been developed [7, 9], yet the general task of retrieval has not been addressed as a full inference task. While we believe this is the most promising of current approaches, we take a different alternative here. Our approach is to give a coherent definition of matching documents based on a probability model, implicitly defining a distance metric.

Another approach is to use dimensionality reduction and perform the query by matching in the lower dimensional space, e.g., see the survey in [10]. A popular method is LSI, a variant of PCA. Instead, we can transfer the approach to multinomial PCA, a statistical model shown to provide better dimensionality reduction [8], as is done here. However, the approach fails in a wide range of cases: suppose the query is Hillary Clinton, and only one document in the collection contain these words. Then there will be no trace of "Hillary Clinton" in the statistical model because it appears in the ignored "noise" components. In this case, something akin to keyword search is essential, or the distance metrics considered above which can introduce the effects of rare words.

A final approach is the query model coupled with a *document language model* [16]. Recent approaches offer additional sophistication [11], but the basic idea is to place a generating distribution on queries that is dependent on a document, but also on characteristics of the corpus as a whole. Search then is to return the document with the maximum likelihood. That is, we develop a distribution

$$p(\text{query} \mid \text{document}, \text{corpus})$$

and then we return the *document* maximizing this likelihood. In principle, this allows us to incorporate all sorts of linguistic and corpus information into the score, however proponents have not done so yet. This distribution is usually built by an *ad hoc* mixture the observed frequency for words in the document, and the observed frequency for words in the entire corpus. Given that the observed word frequencies for a document is a poor model at best (as illustrated by the poor performance of standard probabilistic clustering for text [10]), this approach certainly needs improvement. However, it provides a nice starting point.

3 Practical Considerations

We need to point out that the theory here is a small part of achieving a workable solution, though we believe it is important to achieve intelligent search. Repre-

senting documents as bags of words is crude at best. Due to the presence of things like noun compounds, named entities, etc., it becomes very misleading to place all words in bags without some sort of language processing beforehand. Noun compounds induce string dependencies in data that corrupt statistical models of the bag. Moreover, real internet data is intrinsically noisy: gross spelling errors, punctuation and grammatical errors, acronyms, abbreviations and colloquialisms are not uncommon once one moves beyond the realm of mass-circulation professional publications and into intranet data, especially with newsletters and email content.

In the theory below, we consider the "bag of words" model [1], where a document is represented as a sparse vector of word indices and their occurrence counts. All positional information is lost. In practice, we use a "bag of lexemes," after basic processing to identify noun compounds, reduce words to their lemmata, and in the case of commercial websites, remove easily identified template content. All this uses standard techniques available, and makes a considerable difference to results. More generally, additional linguistic processing should be used to perform tasks such as spelling correction, disambiguation, and anaphora resolution, to clean up the quality of the text, and the "bag of lexemes" needs to be replaced so that analysis and retrieval can deal with local context. Our ideal is represented in the diagram in Fig 1. The role of the current paper is

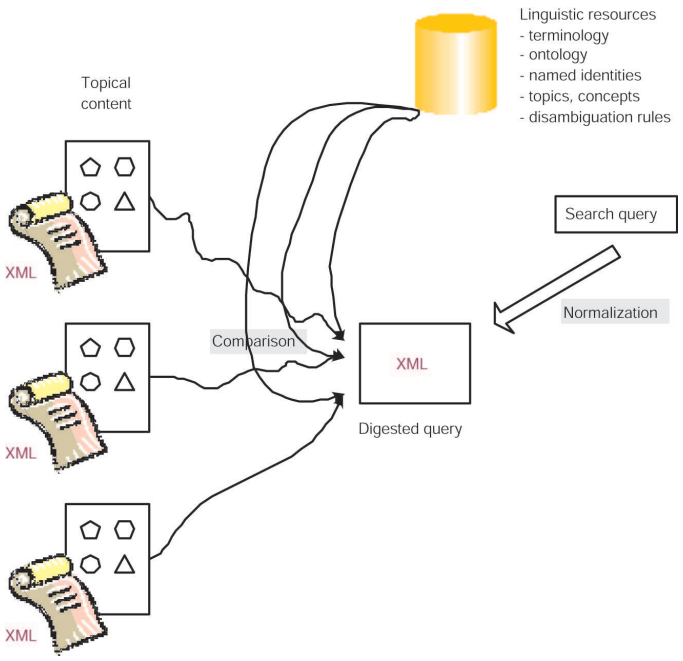


Fig. 1. An ideal XML free-text search function

to demonstrate the improved results one obtains when some small amount of linguistic processing is combined with a non-trivial topic model of documents.

4 Multinomial PCA

This sections introduces our topic model of documents from [3]. This is the multinomial PCA model, which has various other versions in the literature. Methods and models include non-negative matrix factorization [12] (which is a Poisson version), probabilistic latent semantic analysis [8], latent Dirichlet allocation [2], and generative aspect models [14]. A good discussion of the motivation for these techniques can be found in [8], a more sophisticated statistical analysis is by Minka [14]. We do not use Minka's methods here because while we have confirmed they produce higher-fidelity models, we have been unable to scale them appropriately for the large scale models we develop. MPCA in the large has a considerably different behaviour than for the small scale results reported by most authors (see [4]).

The bag of words model is as follows. Ignoring sparsity (which for the subsequent theory is an implementation issue), with J different words the i-th document becomes a vector $\boldsymbol{x}_i \in \mathcal{Z}^J$ where $x_{i,j}$ is the number of occurrences of the j-th word in the i-th document, and where the total $\sum_j x_{i,j}$ gives words in the document. There are I documents in total. Note that the bag-of-words representation is poor when it comes to dealing with the multiple senses of individual words (e.g., the use of "hot" in "hot coffee" and "hot car"), and is thus not very effective when performing semantic analysis of text, but it is a respectable first approximation.

4.1 The Model

A Gaussian model Consider Tipping *et al.*'s representation of standard PCA. A hidden variable $\boldsymbol{m}_i$ is sampled for the i-th document from K-dimensional Gaussian noise. Each entry represents the strength of the corresponding component and can be positive or negative. This is folded with the $K \times J$ matrix of component means $\boldsymbol{\Omega}$ to yield a J-dimensional document mean. Thus for the i-th document from the collection of I in total:

$$\begin{aligned} \boldsymbol{m}_i &\sim \mathit{Gaussian}(0, \mathbf{I_K}) \\ \boldsymbol{x}_i &\sim \mathit{Gaussian}(\boldsymbol{m}_i \boldsymbol{\Omega} + \boldsymbol{\mu}, \mathbf{I_J}\sigma) \end{aligned}$$

This relies on the data being somewhat Gaussian, which fails badly for document data where low and zero counts are normal. A square root transform can make the Poisson-like count data more Gaussian, but it still fails to treat zeros well.

The discrete analogue Modify the above as follows. First sample a probability vector $\boldsymbol{m}_i$ that represents the proportional weighting of components, and then

mix it with a matrix $\boldsymbol{\Omega}$ whose rows represent a word probability vector for a component:

$$\begin{aligned} \boldsymbol{m}_i &\sim Dirichlet(\boldsymbol{\alpha}) \\ \boldsymbol{x}_i &\sim Multinomial(\boldsymbol{m}_i \boldsymbol{\Omega}, L_i) \end{aligned}$$

where L_i is the total number of words in the i-th document, and $\boldsymbol{\alpha}$ is a vector of K-dimensional parameters to the Dirichlet. Thus the probability of the j-th word appearing in the i-th document is a convex combination of the document's hidden vector $\boldsymbol{m}_i$ and the j-th column of $\boldsymbol{\Omega}$. MPCA thus represents an additive/convex mixture of probability vectors.

Note also that a J-dimensional multinomial results from taking J independent Poissons and assuming their total count is given, thus a Poisson or multinomial analysis are similar and we just pursue the one here.

4.2 The Extremes of MPCA

With each document a vector $\boldsymbol{x} \in \mathcal{Z}^J$, traditional clustering becomes the problem of forming a mapping $\mathcal{Z}^J \mapsto \{1, \ldots, K\}$, where K is the number of clusters, whereas dimensionality reduction forms a mapping $\mathcal{Z}^J \mapsto \mathcal{R}^K$ where K is considerably less than J. Instead, MPCA represents the document as a convex combination. This forms a mapping $\mathcal{Z}^J \mapsto \mathcal{C}^K$ where $\mathcal{C}^K$ denotes the subspace of $\mathcal{R}^K$ where every entry is non-negative and the entries sum to 1 ($\boldsymbol{m} \in \mathcal{C}^K$ implies $0 \leq m_k \leq 1$ and $\sum_k m_k = 1$).

Suppose $\boldsymbol{m} \in \mathcal{C}^K$ is the reduction of a particular document. For multi-faceted clustering, $\boldsymbol{m}$ should have most entries zero, and only a few entries significantly depart from zero. For dimensionality reduction, $\boldsymbol{m}$ should have many non-zero entries and many significantly different from zero so that the reduced space $\mathcal{C}^K$ is richly filled out to make the dimensionality reduction efficient in its use of the K dimensions. A measure we shall use for this is entropy, $H(\boldsymbol{m}) = \sum_j m_j \log 1/m_j$. Thus multi-faceted clustering prefers low entropy reductions in $\mathcal{C}^K$ whereas dimensionality reduction prefers high entropy reductions. In the limit, when the average entropy is 0, the mapping becomes equivalent to standard clustering.

5 Theory Review

Here we summarise relevant probability modelling from [3]. A notation convention used here is that indices i, j, k in sums and products always range over 1 to I, J, K respectively, where i denotes a sample index, j a dictionary word index, and k a component index. I is the number of documents, J the number of words (dictionary size), and K the number of components. The index i is usually dropped for brevity (it is shared by all document-wise parameters) and is inserted when needed using the notation "$[i]$".

5.1 A Probability Model

Priors In the MPCA model $\boldsymbol{m}$, a K-dimensional probability vector, represents the proportion of components in a particular document. $\boldsymbol{m}$ must therefore represent quite a wide range of values with a mean representing the general frequency of components in the sample. A prior for such a probability vector is the Dirichlet, $\boldsymbol{m} \sim \text{Dirichlet}(\boldsymbol{\alpha})$. This is analytically attractive but otherwise has little to recommend it. The matrix $\boldsymbol{\Omega}$ represents word frequency probability vectors row-wise. A suitable prior for these is a Dirichlet whose mean correspond to the empirical frequencies of words occurring in the full data set, $\boldsymbol{f}$, so that $\boldsymbol{\Omega}_{k,\cdot} \sim \text{Dirichlet}(2\boldsymbol{f})$ (i.e., an empirical prior).

Likelihoods We briefly present the case where the bag of words has no order. The leads to identical theory to the so-called unigram model, or 0-th order Markov model which retains order. A hidden variable $\boldsymbol{w}$ is used which is a sparse matrix whose entry $w_{k,j}$ is the count of the number of times the j-th word occurs in the document representing the k-th component. Its row total $r_k = \sum_j w_{k,j}$ is the count of the number of words in the document representing the k-th component. Its column total $c_j = \sum_k w_{k,j}$ is the observed data. Denote by $\boldsymbol{w}_{k,\cdot}$ the k-th row vector.

$$\begin{aligned} \boldsymbol{m} &\sim \text{Dirichlet}(\boldsymbol{\alpha}) \\ \boldsymbol{r} &\sim \text{Multinomial}(\boldsymbol{m}, L) \\ \boldsymbol{w}_{k,\cdot} &\sim \text{Multinomial}(\boldsymbol{\Omega}_{k,\cdot}, r_k) \qquad \text{for } k = 1, \ldots, K \end{aligned}$$

The hidden variables here are $\boldsymbol{m}$ and $\boldsymbol{w}$ and the row and column totals are derived. The full likelihood for a single document $p(\boldsymbol{m}, \boldsymbol{w} \mid \boldsymbol{\alpha}, \boldsymbol{\Omega})$ then simplifies to:

$$\frac{1}{Z_D(\boldsymbol{\alpha})} C^L_{w_{1,1}\ldots,w_{K,1},\ldots,w_{K,J}} \prod_k m_k^{\alpha_k - 1} \prod_{k,j} m_k^{w_{k,j}} \Omega_{k,j}^{w_{k,j}} \ , \tag{1}$$

where $Z_D()$ is the normalising constant for a Dirichlet. This is the likelihood used in constructing a variational EM algorithm with the hidden variables. It corresponds to one big multinomial because $\sum_{k,j} m_k \Omega_{k,j} = 1$. Thus the hidden variable $\boldsymbol{w}$ can be marginalized out to yield the formulation in Section 4.

$$\frac{1}{Z_D(\boldsymbol{\alpha})} C^L_{c_1\ldots,c_J} \prod_k m_k^{\alpha_k - 1} \prod_j \left(\sum_j m_k \Omega_{k,j} \right)^{c_j} \ .$$

6 Language Models for Search

The fundamental idea here is that a query is somehow generated by a probabilistic model based on a document. To test whether a document $\boldsymbol{x}$ matches a query, we turn the task around and compute the probability that the query matches the document. This goes as follows, for query $\boldsymbol{q}$ consisting of word

indexes $q_1, q_2, \ldots, q_Q$, we computer the probability for the query given the particular document to match with $\boldsymbol{x}$ and the broader context provided by the MPCA model for the document collection $\boldsymbol{\alpha}, \boldsymbol{\Omega}$. This is broken down with an independence assumption, as in [11].

$$\log p(\boldsymbol{q} \text{ matches} \,|\, \boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\Omega}) \;=\; \sum_{l=1,\ldots,Q} \log p(q_l \text{ matches} \,|\, \boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\Omega})$$

The match for an individual word is based on the notion that each word is either an exact match with one in the document, or a match with a word that could reasonably be generated by the model for the document assigned by MPCA. This distinction is maintained by a hidden indicator variable δ_l with value one for an exact match and zero for a possible match. It has a binomial distribution with parameter ρ. Thus

$$\begin{aligned}\log p(q_l \text{ matches}, \delta_l \,|\, \boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\Omega}, \rho) \;&=\; \rho^{\delta_l}(1-\rho)^{1-\delta_l} \\ &(\delta_l\, p(q_l, \,|\, \boldsymbol{x}, \text{ observed frequency}) + (1-\delta_l)\, p(q_l, \,|\, \boldsymbol{x}, \text{ MPCA model}, \boldsymbol{\alpha}, \boldsymbol{\Omega}))\end{aligned}$$

In this formulation, ρ is a free parameter and is estimated from the document-query combination. Thus

$$\begin{aligned}&\log p(q_l \text{ matches} \,|\, \boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\Omega}, \rho) \;= \\ &\quad \rho\, p(q_l, \,|\, \boldsymbol{x}, \text{ observed frequency}) + (1-\rho)\, p(q_l, \,|\, \boldsymbol{x}, \text{ MPCA model}, \boldsymbol{\alpha}, \boldsymbol{\Omega})\end{aligned}$$

where $p(q_l, \,|\, \boldsymbol{x}, \text{ observed frequency})$ is the observed frequency of the word q_l in the document and $p(q_l, \,|\, \boldsymbol{x}, \text{ MPCA model}, \boldsymbol{\alpha}, \boldsymbol{\Omega})$ is its probability according to the MPCA model for the document $\boldsymbol{x}$. The mixing parameter ρ is set to its average for the query on this document, which is approximately the proportion of exact matches for the query to the document and we return the final score using the approximation

$$p(q_l \text{ matches} \,|\, \boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\Omega}) \;\cong\; p(q_l \text{ matches} \,|\, \boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\Omega}, \hat{\rho})$$

7 Experiments

Our choice of document collection is the new Reuters Corpus[1], which contains 806,791 news items from 1996 and 1997. While there are a number of sizeable web collections available, especially from TREC, and we have our own large Amazon crawls, we chose to use the Reuters collection for these first experiments. Separately, we have been very pleased with the performance of the system on the Amazon crawl (and in fact demonstrate comparisons with Google to industrial partners on this crawl), the Reuters Corpus is a much better controlled baseline. It has few of the noise problems associated with the internet, and is much easier for us to understand, and thus interpret results. Because we want to evaluate

[1] Volume 1: English Language, 1996-08-20 to 1997-08-19.

long 100 word queries as well as shorter ones, the news environment is ideal since many of us can act as experts. Moreover, titles for these documents is usually a good indication of content, thus quick evaluation is feasible.

For our experiments we collected bag-of-words data from the new Reuters Corpus. The average length of a news item is 225 words, which translates to a total of 180 million word instances. About half a million of these are distinct words after the suffix "'s" is eliminated. We use simple statistical tests to identify compounds, together with a compound dictionary extracted from WordNet. We then use a standard fast tagger/lemmatizer system off the internet to represent words to a basic lexeme. Most commonly, this reduces plurals and possessive case. We kept the most frequent $65,000$ lexemes so generated. For instance, "Bill Clinton, Bill Cosby, Bill Gates" are compounds, but "Chelsea Clinton" is not.

Our code is written in C and C++ using Open Source tools and libraries such as the GNU Scientific Library (GSL). The GSL comes with a wide variety of distributional sampling algorithms as well as functions such as the digamma function and its derivatives. More details of some of the techniques used to scale up the MPCA algorithm appear in [4], and to our knowledge we are the only group applying this class of methods to over 50,000 documents at once. The new language models for querying have been implemented in a brute force manner doing a full pass of all documents, and thus is very slow at present. As a comparison, we compared our approach against the standard TF-IDF metric for comparing documents. The TF-IDF implementation uses the inverted index so is fast. Both algorithms throw away query words that are in the most 100 frequent words in the collection (i.e., approximately the stop words).

We evaluated a number of short queries, and a number of long queries. Long queries are text fragments, perhaps 100 words, of interest. These are usually not evaluated in search comparisons, however, we feel they will be a fundamental part of future search systems, computational difficulties not withstanding. The long queries were general news items taken off the internet on 12/6/03 and 13/06/03. These are given below:

1. Iraq war Spain
2. France tourism
3. Aero-engines manufacturer Rolls Royce
4. In a major discovery that may fill in the missing piece connecting us to our most immediate ancestors, the fossilized skulls of two adults and a child who lived in Ethiopia 160000 years ago could represent the earliest known remains.
5. Gregory Peck always seemed to be fighting for good causes. He did so in his most memorable performance as lawyer Atticus Finch in the 1962 film To Kill A Mockingbird and in dozens of other movies from his debut in 1944's Days Of Glory (as a Russian guerrilla) to his last (on TV) in 1993's The Portrait. Last week the American Film Institute named his role in To Kill A Mockingbird as the greatest movie hero of all time. It won Peck his only Oscar although he received four other nominations.

6. On the eve of an expected U.N. vote on the new global criminal court, human rights groups accused the Bush administration of using unconscionable tactics to undermine the tribunal rather than prosecute mass murderers in the 21st century.

Titles of the top twenty articles extracted are presented in the appendix. Evaluation of a short query by our new methods takes a few seconds on our 1.5GHz Pentium Linux machine, and the long queries takes several minutes. For the long queries, our new method is clearly superior, with TF-IDF performing poorly. Our new method is marginally superior in two of the short queries.

8 Conclusion

This is clearly preliminary work in the following sense:

- No attempt has been made to optimize or scale the search algorithm.
- Proper account of difficulties and methods relevant to the internet has not been made, for instance the use of hyper-link information, and linguistic preprocessing to overcome noise in internet text.
- Experiments need to be performed on true Internet data and with a greater variety of state-of-the-art search metrics.

However, the results are surprisingly good. In the larger queries, a service we believe will become critical in intranet contexts (and in the internet context if it could be scaled), our new methods returns results quiet acceptable to users. The quality of TF-IDF in the longer queries could not form the basis of a commercial system in our view. So we see tremendous opportunities here for developing intranet search applications where these kinds of methods should be computationally cost-effective.

Acknowledgements

This work was supported by the Academy of Finland.

References

[1] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, 1999.
[2] D. Blei, A.Y. Ng, and M. Jordan. Latent Dirichlet allocation. In *NIPS*14*, 2002.
[3] W. Buntine. Variational extensions to EM and multinomial PCA. In *ECML 2002*, 2002.
[4] W.L. Buntine and S.Perttu. Is multinomial pca multi-faceted clustering or dimensionality reduction? In C.M. Bishop and B.J. Frey, editors, *Proc. of the Ninth International Workshop on Artificial Intelligence and Statistics*, 2003.
[5] G. Forman. Choose your words carefully: An empirical study of feature selection metrics for text classification. In *PKDD 2002*, 2002.

[6] Moises Goldszmidt and Mehran Sahami. A probabilistic approach to full-text document clustering. Technical Report ITAD-433-MS-98-044, SRI International, 1998.
[7] K. Hall and T. Hofmann. Learning curved multinomial subfamilies for natural language processing and information retrieval. In *ICML 2000*, 2000.
[8] T. Hofmann. Probabilistic latent semantic indexing. In *Research and Development in Information Retrieval*, pages 50–57, 1999.
[9] T. Hofmann. Learning the similarity of documents. In *Advances in Neural Information Processing Systems 12*, pages 914–920, 2000.
[10] G. Karypis and E.-H. Han. Concept indexing: A fast dimensionality reduction algorithm with applications to document retrieval and categorization. In *CIKM-2000*, 2000.
[11] J. Lafferty and C. Zhai. Document language models, query models and risk minimization for information retrieval. In *SIGIR 2001*, New York, 2001.
[12] D. Lee and H. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, 1999.
[13] Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, Massachusetts, 1999.
[14] T. Minka and J. Lafferty. Expectation-propagation for the generative aspect model. In *UAI-2002*, Edmonton, 2002.
[15] D. Moldovan, S. Harabagiu, R. Girju, P Morarescu, N.A. Lacatusu, A. Badulescu, and O. Bolohan. Lcc tools for question answering. In *Proc. 11th TREC*, 2002.
[16] Jay M. Ponte and W. Bruce Croft. A language modeling approach to information retrieval. In *Research and Development in Information Retrieval*, pages 275–281, 1998.

9 Appendix: Results

In this appendix we list the titles of the top 20 documents selected. The new method is on the left, and the TF-IDF results are on the right.

9.1 Iraq war Spain

SPAIN: Iraq to use Syrian port to import aid - Rasheed.
SPAIN: PRESS DIGEST - Spain - Sept 12.
SPAIN: PRESS DIGEST - SPAIN - SEPT 13.
SPAIN: PRESS DIGEST - Spain - Sept 5.
IRAQ: PRESS DIGEST - Iraq - Nov 16.
UK: Iraq asks Spain to help Uday Hussein – opposition.
SPAIN: Spain says wishes U.S. had held off on Iraq attack.
IRAQ: PRESS DIGEST - Iraq - Feb 15.
SPAIN: Repsol to finalise Iraq oilfield deal - Rasheed.
IRAQ: PRESS DIGEST - Iraq - Dec 23.
FRANCE: European allies split over U.S. strike on Iraq.
SPAIN: PRESS DIGEST - Spain - Sept 4.
IRAQ: PRESS DIGEST - Iraq - July 22.
UK: PRESS DIGEST - London-based Arab newspapers - Jan 15.
IRAQ: Saddam's son telephones Iraq soccer team.
SPAIN: Iraq is fully cooperating with U.N. - Rasheed.
UK: Reuters Historical Calendar - September 13.
SPAIN: Spain to resume diplomatic activity in Iraq.
FRANCE: French spy photos put query on US Iraq move - daily.
REPUBLIC OF IRELAND: EU differs over U.S. raid on Iraq - Spring.

FRANCE: European allies split over U.S. strike on Iraq.
USA: Response to U.S. move shows anti-Iraq pact weaker.
SPAIN: PRESS DIGEST - SPAIN - SEPT 13.
REPUBLIC OF IRELAND: EU differs over U.S. raid on Iraq - Spring.
REPUBLIC OF IRELAND: EU differs over U.S. raid on Iraq - Spring.
SPAIN: PRESS DIGEST - Spain - Sept 12.
UK: Reuters Historical Calendar - September 13.
UK: Reuters historical calendar - August 23.
UK: Reuters Historical Calendar - February 15.
INDIA: Oil producers, consumers begin gathering at Goa meet.
UK: Reuters historical calendar - April 14.
UK: REUTERS HISTORICAL CALENDAR - FEBRUARY 22.
UK: UK's Labour promises defence review but no cuts.
UK: Reuters historical calendar - August 18.
UK: Reuters historical calendar - December 6.
FRANCE: France, Britain at odds over Zaire at summit.
UK: Reuters historical calendar - June 7.
UK: Reuters historical calendar - September 25.
UK: Reuters historical calendar - May 30.
UK: Reuters historical calendar - July 24.

Note the historical calendars retrieved by TF-IDF list important dates in the British Empire going back to the 1800's when Britian controlled Iraq. The

Spanish and Iraqi press digests list diplomatic and other exchanges between the countries relating to war and U.N. issues.

9.2 France tourism

FRANCE: Club Med sells stakes in Valtur, Situr.
MOROCCO: Moroccan tourism picks up in first seven months.
FRANCE: FRANCE-TELEVISION-MEDIAS-CANAL-ABONNEMENTS.
AUSTRALIA: Australia, Canada and French TV bodies in pact.
TUNISIA: Tunisian expatriates remit 754 mln dinars in 1996.
FRANCE: Algeria's Zeralda tourism firm loses 17 mln dinars.
MALTA: Malta tourism down 5.6 percent.
MALI: Impoverished Mali plugs into the Internet.
FRANCE: CYCLING-TOUR DE FRANCE TO START FROM IRELAND IN 1998.
FRANCE: GAUMONT-RESULTATS.
AUSTRALIA: RTRS-BRL Hardy joins with French wine-maker.
AUSTRALIA: TENNIS-RESULTS OF SYDNEY INTERNATIONAL TENNIS TOURNAMENT.
SPAIN: France most visited country in 1996 - study.
SPAIN: France most visited destination in 1996 - study.
FRANCE: Pathe buys TV channel from Landmark.
UK: International tourism seen growing through yr 2000.
BELGIUM: International anti-drugs swoop nets 123 people.
FRANCE: France to crack down on child sex.
SOUTH AFRICA: France bids to boost S.Africa sales, big projects.
TUNISIA: PRESS DIGEST - Tunisia - Jan 18.
CUBA: Cuba's tourism arrivals, income grow in 1996.
GERMANY: World tourism fair opens with environment worries.
MALTA: Malta tourism down 5.6 percent.
SPAIN: France most visited country in 1996 - study.
MOROCCO: Moroccan tourism picks up in first seven months.
SPAIN: France most visited destination in 1996 - study.
UK: International tourism seen growing through yr 2000.
CUBA: Despite U.S. embargo, Cuban tourism climbs.
MALTA: Malta tourism dips but appears to be recovering.
FRANCE: MEPs demand urgent clampdown on sex tourism.
SYRIA: 2.4 million tourists visited Syria in 1996.
ROMANIA: ROMANIA TO PRIVATISE HOTELS TO REJUVENATE TOURISM.
ROMANIA: Romania to privatise hotels to rejuvenate tourism.
SOUTH AFRICA: France bids to boost S.Africa sales, big projects.
MALTA: Malta tourism drops 5.3 percent in February.
MAURITIUS: Mauritius tourist arrivals up 17 pct in Q1 1997.
USA: International tourists to U.S. at all-time high.
FRANCE: Commission drafts measures to fight child sex tourism.
FRANCE: CYCLING-TOUR DE FRANCE TO START FROM IRELAND IN 1998.
FRANCE: EU prepares measures to fight child sex tourism.

9.3 Aero-engines manufacturer Rolls Royce

UK: Rolls Royce soars on aero engines hopes.
UK: Rolls Royce admits engine failures - paper.
UK: OPTIONS -Straddle sold in Rolls Royce.
UK: Rolls Royce says to sell Bristol Aerospace.
UK: OPTIONS - Straddles traded in British Telecom.
UK: OPTIONS - Straddles traded in Abbey National.
UK: OPTIONS - Volatility trades struck in UK equities.
UK: OPTIONS-B.A.T puts, calls sold for premium.
UK: OPTIONS - Short position rolled forward in BP.
UK: Air Canada to fly aero-engine to Boeing.
UK: Rolls Royce signs China training agreement.
UK: PRESS DIGEST - Financial Times - Sept 2.
UK: FOCUS - Britain commits to Eurofighter at air show.
FRANCE: Aero-engine firms see more exclusive sale pacts.
FRANCE: Monarch orders four more Airbus craft.
SINGAPORE: Rolls-Royce to raise engine profile in Asia.
UK: Rolls Royce wins $55 mln BA order.
UK: OPTIONS -CALL SPREAD SOLD IN ICI.
UK: Rolls Royce gets $500 mln Trent 700 order.
FRANCE: FOCUS-AMR to buy Brazil jets in $1 bln order.
UK: Rolls Royce soars on aero engines hopes.
UK: Rolls Royce says to sell Bristol Aerospace.
UK: OPTIONS -Straddle sold in Rolls Royce.
UK: Rolls Royce admits engine failures - paper.
UK: OPTIONS - Straddles traded in Abbey National.
UK: OPTIONS - Volatility trades struck in UK equities.
UK: OPTIONS - Straddles traded in British Telecom.
UK: OPTIONS-UK August stock options busy in thin trade.
UK: OPTIONS-B.A.T puts, calls sold for premium.
UK: OPTIONS - Short position rolled forward in BP.
UK: PRESS DIGEST - Financial Times - Sept 2.
UK: Rolls Royce signs China training agreement.
UK: Air Canada to fly aero-engine to Boeing.
UK: OPTIONS -CALL SPREAD SOLD IN ICI.
UK: FOCUS - Britain commits to Eurofighter at air show.
SINGAPORE: Rolls-Royce to raise engine profile in Asia.
FRANCE: Aero-engine firms see more exclusive sale pacts.
UK: Rolls says close to sale of steam power units.
UK: Rolls-Royce shares up as dividend raised.
UK: Steam power hits Rolls-Royce year profits.

9.4 Major discovery

In a major discovery that may fill in the missing piece connecting us to our most immediate ancestors, the fossilized skulls of two adults and a child who lived in Ethiopia 160000 years ago could represent the earliest known remains.

UK: We may be older than we think, scientist says.
UK: Oldest-ever stone tools found in Africa.
CANADA: FEATURE-Pow wow a return to Canadian Indian tradition.
RWANDA: FEATURE - School is Rwanda's grisly memorial to genocide.
SPAIN: Spanish scientists find possible new human species.
BANGLADESH: Bangladeshi men castrated by jealous wives.
BANGLADESH: FEATURE - Bangladesh struggles to end flesh trade.
UK: UK researchers find biological link to anorexia.
UK: Scientists trace 9,000-yr-old skeleton's kin.
MOZAMBIQUE: Thieves leave Mozambican airport in the dark.
SIERRA LEONE: Slave's song brings African American grandma home.
USA: Bones may be of Florida teens missing 18 years.
AUSTRALIA: RTRS-TIMELINES-Today in History - Dec 24.
UK: Aboriginal tells Britain to return ancestor's head.
CHAD: FEATURE - Blind Chadian teenage pianist is symbol of hope.
USA: Brazil insect helps protect African cassava crops.
GREECE: FEATURE - Philip of Macedon's tomb reveals secrets.
AUSTRALIA: YEAREND - Misfits with grudges were year's worst killers.
USA: Funeral for six-year-old strangled in Colorado home.
BHUTAN: FEATURE - Sleepy Bhutan awakens to tourism.
SOUTH AFRICA: Revered skull of S.Africa king is Scottish woman's.
EGYPT: Egypt experts dig up cancer in ancient skull.
BELGIUM: Police sound final whistle on skull kick-about.
ETHIOPIA: Donors pledge $2.5 billion for Ethiopia.
EGYPT: Lonely Egypt widow digs ups dead husband's skull.
ETHIOPIA: Ethiopia economy set to grow, central banker says.
ETHIOPIA: Ethiopia farm production not affected by drought.
ETHIOPIA: Ethiopia seeks to renegotiate Russian loans.
JAPAN: Whales, hippos, cows share ancestor - Japan research.
AUSTRALIA: RTRS-Broker crosses 10 pct of Discovery.
ETHIOPIA: Ethiopia says financial sector underdeveloped.
UK: Scientists push back date of earliest humans' era.
AUSTRALIA: RTRS-Discovery says forms new oil alliance.
ETHIOPIA: Ethiopia achieved record cereal harvest in 1995.
ETHIOPIA: Ethiopia says sold $1 bln at auctions in 96/97.
ETHIOPIA: Ethiopia names new agriculture minister.
ETHIOPIA: Ethiopia appeals for $2 billion in foreign aid.
ETHIOPIA: Ethiopia's justice minister resigns.
AUSTRALIA: RTRS-Premier plans talks with Discovery.
AUSTRALIA: RTRS - Discovery rejects Premier bid.

9.5 Gregory Peck

Gregory Peck always seemed to be fighting for good causes. He did so in his most memorable performance as lawyer Atticus Finch in the 1962 film To Kill A Mockingbird and in dozens of other movies from his debut in 1944's Days Of Glory (as a Russian guerrilla) to his last (on TV) in 1993's The Portrait. Last week the American Film Institute named his role in To Kill A Mockingbird as the greatest movie hero of all time. It won Peck his only Oscar although he received four other nominations.

ITALY: Venice Film Festival embraces art in all forms.
ITALY: Venice Film Festival embraces art in all forms.
USA: Actress Marjorie Reynolds dies aged 77.
USA: "The English Patient" takes lion's share of Oscars.
USA: Films in Robert Mitchum's career.
CZECH REPUBLIC: Czech Oscar winners to make film about Britain's RAF.
FRANCE: Star-studded gala, Wenders film mark Cannes 50th.
FRANCE: France revels in Binoche's surprise Oscar win.
FRANCE: Star-studded gala, Wenders film mark Cannes' 50th.
USA: Maine library tracks celebrity reading.
ITALY: Venice prize goes to IRA film "Michael Collins".
FRANCE: French film, theatre actress Casares dies.
MEXICO: France honours "rebel beauty" of Mexico screen goddess.
CZECH REPUBLIC: CZECHS GIVE OSCAR WINNERS BEER TOAST IN HOMECOMING.
USA: Legendary director Fred Zinnemann dies at 89.
CANADA: Quebec movie director and actress die in plane crash.
UK: Wilde the new Oscar winner in Britain.
FRANCE: French heap praise on Italian actor Mastroianni.
FRANCE: Egypt director preaches tolerance after film ban.
FRANCE: France revels in Binoche's win.
UK: Briton sues over televised suicide bid.
USA: American Mobile Satellite names CFO.
USA: Cyberian Outpost shuts real doors as online sales soar.
NETHERLANDS: INTERVIEW-POLYGRAM SAYS FILM PLAN ON TRACK.
NETHERLANDS: INTERVIEW-PolyGram says film plan on track.
USA: N.Y. official sees '97 film, TV production gains.
USA: Peck Comm. Schools, Mich., Aa - Moody's.
TAIWAN: China director, movie sweep Taiwan's "Oscars".
INDONESIA: Indonesia's film hopes pinned on blockbuster.
USA: Oldest complete U.S. movie found in Oregon basement.
USA: Hollywood's independent studios not so independent.
USA: Hollywood's independent studios not so independent.
INDONESIA: FEATURE-Indonesian film industry hopes pinned on blockbuster.
USA: Peck Comm Sch Dist, Mich. won by Wm. Hough.
USA: FEATURE-Hollywood writers lament sorry state of film.
USA: Oscar nominations for best foreign film.
GERMANY: German film feted as opening act of Berlin fest.
UK: Film director Zinnemann dies of heart attack.
GERMANY: Global film buyers look for hits at Berlin market.
GERMANY: Global film buyers look for hits at Berlin market.

9.6 New global criminal court

On the eve of an expected U.N. vote on the new global criminal court, human rights groups accused the Bush administration of using unconscionable tactics

to undermine the tribunal rather than prosecute mass murderers in the 21st century.

UNITED NATIONS: U.S. urges U.N. assembly action on human rights.
UNITED NATIONS: U.S. INSISTS ON U.N. COUNCIL ROLE IN CRIMINAL COURT.
UNITED NATIONS: Main issues in establishing world criminal court.
USA: U.S. human rights group criticizes major powers.
AUSTRIA: Austria calls for human rights action plan.
SWITZERLAND: US criticises Chinese tactics at UN rights forum.
USA: Clinton puts political spin on human rights speech.
UNITED NATIONS: UN rebukes Burma for continued political supression.
UNITED NATIONS: Albright leads U.N. criticism of Burma government.
USA: Group charges powers with human rights hypocrisy.
SWITZERLAND: World jurists body says Tunisia seized lawyer.
VATICAN: Vatican demands West help end world hunger.
switzerland: Amnesty slams UN on China, Turkey, Algeria abuses.
SWITZERLAND: RIGHTS-CHINA-DILEMMA (NEWS ANALYSIS, SCHEDULED).
UNITED NATIONS: Clinton signs test ban treaty, urges further steps.
ITALY: Gaddafi sees more Islamic attacks in U.S. - paper.
CAMBODIA: Cambodia's new drug law comes under fire.
UK: Monthly Review - Big refugee exodus from Zaire.
MALI: Christopher campaigns for democracy on Africa tour.
NORWAY: China dissident Wei nominated for 1997 Peace Prize.
UNITED NATIONS: Prospects grow for an international criminal court.
NETHERLANDS: Funds row forces U.S. lawyers out of U.N. tribunal.
NETHERLANDS: Plans for world criminal court held up by haggling.
USA: VarTech acquires 21st Century Professional.
TANZANIA: U.N. Rwanda tribunal head defends his record.
UNITED NATIONS: U.N. moves to clean up Rwanda tribunal structures.
TANZANIA: U.N. Rwanda genocide tribunal receives key accused.
RWANDA: Rwanda says U.N. genocide tribunal useless.
TANZANIA: UN investigates Rwanda genocide tribunal.
SOUTH KOREA: S.Korea seeks open market in 21st century.
KENYA: Rwanda genocide tribunal employee murdered.
NETHERLANDS: U.N. tribunal soldiers on despite Bosnia inertia.
NETHERLANDS: U.N. tribunal judges criticise Bosnia conference.
SWITZERLAND: Swiss court rules Rwandan may face U.N. tribunal.
SWITZERLAND: US criticises Chinese tactics at UN rights forum.
YUGOSLAVIA: Yugoslavia lets UN war-crimes tribunal open office.
SWITZERLAND: Amnesty slams U.N. human rights forum.
TANZANIA: Defence says genocide tribunal has no jurisdiction.
NETHERLANDS: War crimes tribunal on ex-Yugoslavia - key facts.
NETHERLANDS: Albright to visit U.N. war crimes tribunal.

Evaluation and Validation of Two Approaches to User Profiling

F. Esposito, G. Semeraro, S. Ferilli, M. Degemmis,
N. Di Mauro, T.M.A. Basile, and P. Lops

Dipartimento di Informatica
Università di Bari
via E. Orabona, 4 - 70125 Bari - Italia
{esposito,semeraro,ferilli,degemmis,nicodimauro,basile,lops}@di.uniba.it

Abstract. In the Internet era, huge amounts of data are available to everybody, in every place and at any moment. Searching for relevant information can be overwhelming, thus contributing to the user's sense of information overload. Building systems for assisting users in this task is often complicated by the difficulty in articulating user interests in a structured form - a profile - to be used for searching. Machine learning methods offer a promising approach to solve this problem. Our research focuses on supervised methods for learning user profiles which are predictively accurate and comprehensible.
The main goal of this paper is the comparison of two different approaches for inducing user profiles, respectively based on Inductive Logic Programming (ILP) and probabilistic methods. An experimental session has been carried out to compare the effectiveness of these methods in terms of classification accuracy, learning and classification time, when coping with the task of learning profiles from textual book descriptions rated by real users according to their tastes.

1 Introduction

The ever increasing popularity of the Internet has led to a huge increase in the number of Web sites and in the volume of available on-line data. Users are swamped with information and have difficulty in separating relevant from irrelevant information. This leads to a clear demand for automated methods able to support users in searching the extremely large Web repositories in order to retrieve relevant information with respect to users' individual preferences. The problem complexity could be lowered by the automatic construction of machine processable profiles that can be exploited to deliver *personalized* content to the user, fitting his or her personal interests.

Personalization has become a critical aspect in many popular domains such as e-commerce, where a user explicitly wants the site to store information such as preferences about himself or herself and to use this information to make recommendations. Exploiting the underlying one-to-one marketing paradigm is essential to be successful in the increasingly competitive Internet marketplace.

Recent research on intelligent information access and recommender systems has focused on the content-based information recommendation paradigm: it requires textual descriptions of the items to be recommended [3].

In general, a content-based system analyzes a set of documents rated by an individual user and exploits the content of these documents to infer a model or profile that can be used to recommend additional items of interest.

In this paper we present a comparison between two different learning strategies to infer models of users' interests from text: an ILP approach and a naïve bayes method. Motivation behind our research is the realization that user profiling and machine learning techniques can be used to tackle the *relevant information problem* already described. Our experiments evaluated the effects of the two above mentioned methods in learning intelligible profiles of users' interests. The experiments were conducted in the context of a content-based profiling system for virtual bookshop on the World Wide Web. In this scenario, a client side utility has been developed in order to download documents (book descriptions) for a user from the Web and to capture users feedback regarding his liking/disliking on the downloaded documents. Then this knowledge can be exploited by the two different machine learning techniques so that when a trained system encounters a new document it can intelligently infer whether this new document will be liked by the user or not. This strategy can be used to make recommendations to the user about new books. The experiments reported here investigate also the effect of using different representations of the profiles.

The structure of the remainder of the paper is as follows: Section 2 describes the ILP system INTHELEX and its main features, while the next section introduces Item Recommender, the system that implements a statistical learning process to induce profiles from text. Then a detailed description of the experiments is given in Section 4, along with an analysis of the experimental results by means of a statistical test. Finally, Section 5 draws some general conclusions.

2 INTHELEX

INTHELEX (INcremental THEory Learner from EXamples) [2] is a learning system for the induction of hierarchical theories from positive and negative examples which focuses the search for refinements by exploiting the Object Identity [8] bias on the generalization model (according to which terms denoted by different names must be distinct). It is fully and inherently incremental: this means that, in addition to the possibility of taking as input a previously generated version of the theory, learning can also start from an empty theory and from the first available example; moreover, at any moment the theory is guaranteed to be correct with respect to all of the examples encountered thus far. This is a fundamental issue, since in many cases deep knowledge about the world is not available. Incremental learning is necessary when either incomplete information is available at the time of initial theory generation, or the nature of the concepts evolves dynamically, which are unnegligible issues for learning user profiles. INTHELEX can learn simultaneously various concepts, possibly related to each other, and

is based on a closed loop architecture — i.e. the learned theory correctness is checked on any new example and, in case of failure, a revision process is activated on it, in order to restore completeness and consistency.

INTHELEX learns theories expressed as sets of Datalog^{OI} clauses (function free clauses to be interpreted according to the Object Identity assumption). It adopts a full memory storage strategy — i.e., it retains all the available examples, thus the learned theories are guaranteed to be valid on the whole set of known examples — and it incorporates two inductive operators, one for generalizing definitions that reject positive examples, and the other for specializing definitions that explain negative examples. Both these operators, when applied, change the set of examples the theory accounts for.

A set of examples of the concepts to be learned is provided by an *Expert*, possibly selected from the *Environment*. Examples are definite ground Horn clauses, whose body describes the observation by means of only basic non-negated predicates of the representation language adopted for the problem at hand, and whose head lists all the classes for which the observed object is a positive example and all those for which it is a negative one (in this case the class is negated). Single classifications are processed separately, in the order they appear in the list, so that the teacher can still decide which concepts should be taken into account first and which should be taken into account later. It is important to note that a positive example for a concept is not considered as a negative example for all the other concepts (unless it is explicitly stated).

The whole set of examples can be subdivided into *tuning* and *test* examples, according to the way in which examples are exploited during the learning process. Specifically, tuning examples, previously classified by the Expert, are exploited to build/refine a theory that is able to explain them. An initial theory can also be provided by the Expert. Subsequently, such a theory, plus the Background Knowledge (if any), is checked against test examples and, in case of incorrectness, the cause of the wrong decision can be located. Test examples are exploited only to check the predictive capabilities of the theory on new observations. Conversely, tuning examples are exploited incrementally to modify incorrect hypotheses according to a data-driven strategy. In particular, when a positive example is not covered, a revised theory is obtained in one of the following ways (listed by decreasing priority) such that completeness is restored:

- replacing a clause in the theory with one of its generalizations against the problematic example;
- adding a new clause to the theory, obtained by properly turning constants into variables in the problematic example;
- adding the problematic example as a positive exception.

When, on the other hand, a negative example is covered, the system outputs a revised theory that restores consistency by performing one of the following actions (by decreasing priority):

- adding positive literals that are able to characterize all the past positive examples of the concept (and exclude the problematic one) to one of the clauses that concur to the example coverage;

– adding a negative literal that is able to discriminate the problematic example from all the past positive ones to the clause in the theory by which the problematic example is covered;
– adding the problematic example as a negative exception.

An exception contains a specific reference to the observation it represents, as it occurs in the tuning set; new incoming observations are always checked with respect to the exceptions before the rules of the related concept. This does not lead to rules which do not cover any example, since exceptions refer to specific objects, while rules contain variables, so they are still applicable to other objects than those in the exceptions.

It is worth noting that INTHELEX never rejects examples, but always refines the theory. Moreover, it does not need to know *a priori* what is the whole set of concepts to be learned, but it learns a new concept as soon as examples about it are available.

We were led by a twofold motivation to exploit INTHELEX on the problem of learning user profiles. First, its representation language (First-Order Logic) is more suitable than numeric/probabilistic approaches to obtain intuitive and human readable rules, which are a highly desiberable feature in order to understand the user preferences. Second, incrementality is an unnegligible requirement in the given task, since new information on a user is available each time he issues a query, and it would be desirable to be able to refine the previously generated profile instead of completely rejecting it and learning a new one from scratch. Moreover, a user's interests and preferences might change in time, a problem that only incremental systems are able to tackle.

Since INTHELEX is not currently able to handle numeric values, it was not possible to learn preference rates in the continuous interval $[0, 1]$ like in the probabilistic approach. Thus, a discretization was needed. Instead of learning a definition for each of the 10 possible votes, we decided to learn just two possible classes of interest: "likes", describing that the user likes a book, and its opposite "not(likes)". Specifically, the former (positive examples) encompasses all rates ranging from 6 to 10, while the latter (negative examples) included all the others (from 1 to 5). It is worth noting that such a discretization step is not in charge of the human supervisor, since a proper abstraction operator embedded in INTHELEX can be exploited for carrying out this task. Moreover, it has a negligible computational cost, since each numeric value is immediately mapped onto the corresponding discretized symbolic value.

Each book description is represented in terms of three components by using predicates `slot_title(b,t)`, `slot_author(b,au)`, and `slot_annotation(b,an)`, indicating that the objects `t`, `au` and `an` are, respectively, the title, author and annotation of the book `b`. Any word in the book description is represented by a predicate corresponding to its stem, and linked to both the book itself and the single slots in which it appears. For instance, predicate `prolog(slott, slottitleprolog)` indicates that the object `slottitleprolog` has stem "prolog" and is contained in slot `slott`; in such a case, also a literal `prolog(book)` is present to say that stem "prolog" is present in the book description.

Also the number of occurrences of each word in each slot was represented by means of the following predicates: `occ_1`, `occ_2`, `occ_m`, `occ_12`, `occ_2m`. A predicate `occ_X(Y)` indicates that term Y occurs X times, while a predicate `occ_XY(Z)` indicate that the term Z occurs from X to Y times. Again, such a 'discretization' was needed because numeric values cannot be dealt with in INTHELEX. Note that all the predicates representing intervals to which the value to be represented belongs must be used to represent it; thus, many such predicates can be needed to represent the occurrences of a term. For instance, if a term occurs once, then it occurs also from 1 to 2 (`occ_12`) times and from 1 to m (`occ_1m`) times. Figure 1 shows an example for the class `likes`. Given the specific value in the example, all the intervals to which it belongs are automatically added by the system by putting this information in the background knowledge and exploting its *saturation* operator.

```
likes(501477998) :-
     slot_title(501477998, slott),
     practic(slott, slottitlepractic),
     occ_1(slottitlepractic),
     occ_12(slottitlepractic),
     prolog(slott, slottitleprolog),
     occ_1(slottitleprolog),
     occ_12(slottitleprolog),
     slot_authors(501477998, slotau),
     l_sterling(slotau, slotauthorsl_sterling),
     occ_1(slotauthorsl_sterling),
     occ_12(slotauthorsl_sterling),
     slot_annotation(501477998, slotan),
     l_sterling(501477998),
     practic(501477998),
     prolog(501477998).
```

Fig. 1. First-Order Representation of a Book

3 Item Recommender

ITR (ITem Recommender) [1] is a system able to recommend items based on their textual descriptions. It implements a probabilistic learning algorithm to classify texts, the naïve Bayes classifier. Naïve Bayes has been shown to perform competitively with more complex algorithms and has become an increasingly popular algorithm in text classification applications [6, 4].

The prototype is able to classify text belonging to a specific category as interesting or uninteresting for a particular user. For example, the system could learn the target concept "*textual descriptions the user finds interesting in the category Computer and Internet*".

Bayesian reasoning provides a probabilistic approach to inference. It is based on the assumption that the quantities of interest are governed by probabilistic distributions and that optimal decision can be made by reasoning about these probabilities together with observed data.

In the learning problem, each instance (item) is represented by a set of *slots.* Each slot is a textual field corresponding to a specific feature of an item.

The text in each slot is a collection of words (a bag of word, BOW) processed taking into account their occurrences in the original text. Thus, each instance is represented as a vector of BOWs, one for each slot.

Moreover, each instance is labelled with a discrete rating (from 1 to 10) provided by a user, according to his or her degree of interest in the item.

According to the Bayesian approach to classify natural language text documents, given a set of classes $C = \{c_1, c_2, \ldots, c_{|C|}\}$, the conditional probability of a class c_j given a document d is calculated as follows:

$$P(c_j|d) = \frac{P(c_j)}{P(d)} P(d|c_j)$$

In our problem, we have only 2 classes: c_+ represents the positive class (user-likes, corresponding to ratings from 6 to 10), and c_- the negative one (user-dislikes, ratings from 1 to 5). Since instances are represented as a vector of documents, (one for each BOW), and assumed that the probability of each word is independent of the word's context and position, the conditional probability of a category c_j given an instance d_i is computed using the formula:

$$P(c_j|d_i) = \frac{P(c_j)}{P(d_i)} \prod_{m=1}^{|S|} \prod_{k=1}^{|b_{im}|} P(t_k|c_j, s_m)^{n_{kim}} \tag{1}$$

where $S = \{s_1, s_2, \ldots, s_{|S|}\}$ is the set of slots, b_{im} is the BOW in the slot s_m of the instance d_i, n_{kim} is the number of occurrences of the token t_k in b_{im}.

In (1), since for any given document, the prior $P(d_i)$ is a constant, this factor can be ignored if the only interest concerns a ranking rather than a probability estimate. To calculate (1), we only need to estimate the probability terms $P(c_j)$ and $P(t_k|c_j, s_m)$, from the training set, where each instance is weighted according to the user rating r:

$$w_+^i = \frac{r-1}{9}; \qquad w_-^i = 1 - w_+^i \tag{2}$$

The weights in (2) are used for weighting the occurrence of a word in a document. For example, if a word appears n times in a document d_i, it is counted as occurring $n \cdot w_+^i$ in a positive example and $n \cdot w_-^i$ in a negative example. Weights are used for estimating the two probability terms according to the following equations:

$$\hat{P}(c_j) = \frac{\sum_{i=1}^{|TR|} w_j^i}{|TR|} \tag{3}$$

$$\hat{P}(t_k|c_j, s_m) = \frac{\sum_{i=1}^{|TR|} w_j^i n_{kim}}{\sum_{i=1}^{|TR|} w_j^i |b_{im}|} \tag{4}$$

In (4), n_{kim} is the number of occurrences of the term t_k in the slot s_m of the i^{th} instance, and the denominator denotes the total weighted length of the slot s_m in the class c_j. Therefore, $\hat{P}(t_k|c_j, s_m)$ is calculated as a ratio between the weighted occurrences of the term t_k in slot s_m of class c_j and the total weighted length of the slot.

The final outcome of the learning process is a probabilistic model used to classify a new instance in the class c_+ or c_-. The model can be used to build a personal profile including those words that turn out to be most indicative of the user's preferences, according to the value of the conditional probabilities in (4).

In the specific context of book recommendations, instances in the learning process are the book descriptions. ITR represents each instance as a vector of three BOWs, one BOW for each slot. The slots used are: *title*, *authors* and *textual annotation*. Each book description is analyzed by a simple pattern-matcher that extracts the words, the *tokens* to fill each slot. Tokens are obtained by eliminating stopwords and applying stemming. Instances are used to train the system: cccurrences of terms are used to estimates probabilities as described in Equations (3) and (4). An example ITR profile is given in figure 2.

4 Experimental Sessions

In this section we describe results from experiments using a collection of textual book descriptions rated by real users according to their tastes. The goal of the experiment has been the comparison of the methods implemented by INTHELEX and ITR in terms of classification accuracy, learning and classification time, when coping with the task of learning user profiles.

4.1 Design of the experiments

Eight book categories were selected at the Web site of a virtual bookshop. For each book category, a set of book descriptions was obtained by analyzing Web pages using an automated extractor and stored in a local database. Table 1 describes the extracted information. For each category we considered:

– *Book descriptions* - number of books extracted from the Web site belonging to the specific category;

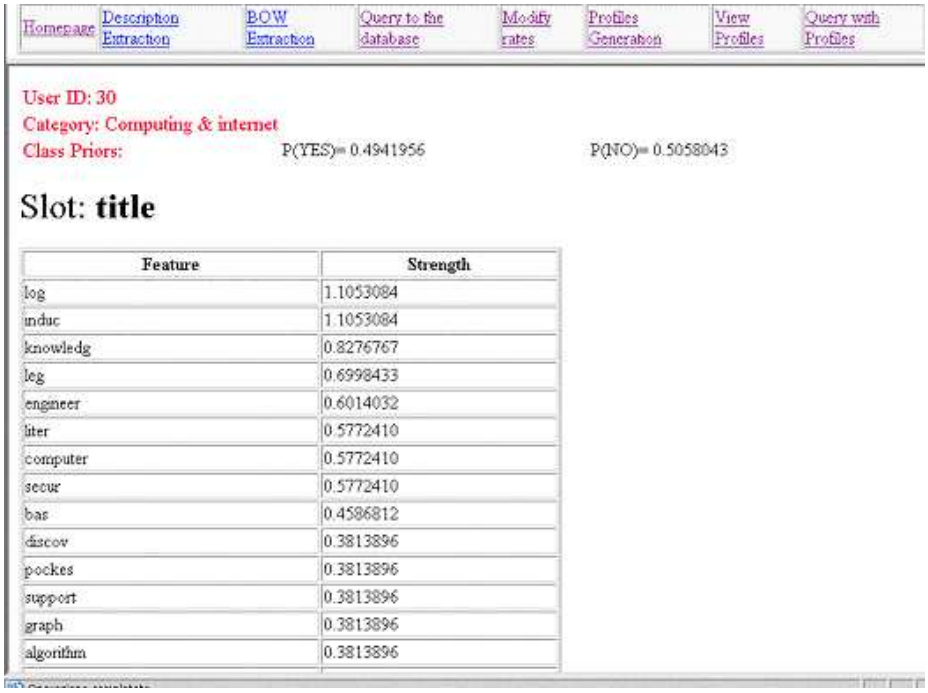


Fig. 2. An example of ITR user profile

- *Books with annotation* - number of books with a textual annotation (slot annotation not empty);
- *Avg. annotation length* - average length (in words) of the annotations;

Several users have been involved in the experiments: each user were requested to choose one or more categories of interest and to rate 40 or 80 books (in the database) in each selected category, providing 1-10 discrete ratings. In this way, for each user a dataset of 40 or 80 rated books was obtained (see Table 2).

On each dataset a 10-fold cross-validation was run and several metrics were used in the testing phase. In the evaluation phase, the concept of *relevant book* is central. A book in a specific category is considered as relevant by a user if his or her rating is greater than 5. This corresponds in ITR to having $P(c_+|d_i) \geq 0.5$, calculated as in equation (1), where d_i is a book in a specific category. Simmetrically, INTHELEX considers as relevant books covered by the inferred theory. Classification effectiveness is measured in terms of the classical Information Retrieval (IR) notions of *precision* (Pr), *recall* (Re) and *accuracy* (Acc), adapted to the case of text categorization [7]. *Precision* is the proportion of items classified as relevant that are really relevant, and *recall* is the proportion of relevant items that are classified as relevant; *accuracy* is the proportion of items that are correctly classified as relevant or not.

Table 1. Database information

Category	Book descr.	Books with annotation	Avg. annotation length
Computing & Int.	5378	4178 (77%)	42.35
Fiction & lit.	5857	3347 (57%)	35.71
Travel	3109	1522 (48%)	28.51
Business	5144	3631 (70%)	41.77
SF, horror & fan.	556	433 (77%)	22.49
Art & entert.	1658	1072 (64%)	47.17
Sport & leisure	895	166 (18%)	29.46
History	140	82 (58%)	45.47
Total	22785	14466	

Table 2. Number of books rated by each user in a given category

UserID	**Category**	**Rated books**
37	SF, Horror & Fantasy	40
26	SF, Horror & Fantasy	80
30	Computer & Internet	80
35	Business	80
24c	Computer & Internet	80
36	Fiction & literature	40
24f	Fiction & literature	40
33	Sport & leisure	80
34	Fiction & literature	80
23	Fiction & literature	40

Table 3. Performance for ITR and INTHELEX on 10 different users

	Precision		**Recall**		**Accuracy**	
UID	ITR	INTHELEX	ITR	INTHELEX	ITR	INTHELEX
37	0,767	0,967	0,883	0,5	0,731	0,695
26	0,818	0,955	0,735	0,645	0,737	0,768
30	0,608	0,583	0,600	0,125	0,587	0,488
35	0,651	0,767	0,800	0,234	0,725	0,662
24c	0,586	0,597	0,867	0,383	0,699	0,599
36	0,783	0,9	0,783	0,3	0,700	0,513
24f	0,785	0,9	0,650	0,35	0,651	0,535
33	0,683	0,75	0,808	0,308	0,730	0,659
34	0,608	0,883	0,490	0,255	0,559	0,564
23	0,500	0,975	0,130	0,9	0,153	0,875
Mean	0,679	0,828	0,675	0,4	0,627	0,636
	(0,699)	(0,811)	(0,735)	(0,344)	(0,68)	(0,609)

Table 4. Learning and Classification times (msec) for ITR and INTHELEX on 10 different users

	Learning Time		**Classification Time**	
UID	ITR	INTHELEX	ITR	INTHELEX
37	3,738	3931,0	0,851	15,0
26	5,378	8839,0	0,969	20,0
30	8,561	51557,0	1,328	53,0
35	9,289	30338,0	1,423	55,0
24c	7,502	29780,0	1,208	44,0
36	5,051	12317,0	0,894	19,0
24f	4,532	18448,0	0,848	19,0
33	5,820	14482,0	0,961	25,0
34	7,592	73708,0	1,209	42,0
23	4,951	1859,0	0,845	20,0
Mean	6,2414	24525,9	1,0536	31,2

As regards training and classification times, we tested the algorithms on a 2.4 GHz Pentium IV running Windows 2000.

4.2 Discussion

Table 3 shows the average precision, recall and accuracy of the models learned in the 10 folds for each user. The last row reports the mean values, averaged on all users. Since the average performance for ITR is very low for user 23, we decided to have a deeper insight into the corresponding training file, and noted that all examples were positive, thus indicating possible noise in the data. This led us to recompute the metrics neglecting this user, thus obtaining the results reported in parentheses.

In general, INTHELEX provides some performance improvement over ITR. In particular, it can be noticed that INTHELEX produces very high precision

even on the category "SF, horror & fantasy", taking into account the shortness of the annotations provided for books belonging to this category. This result is obtained both for user 26, who rated 80 books, and for user 37, who rated only 40 books. Moreover, classification accuracy obtained by INTHELEX is slightly better than the one reached by ITR. On the other hand, ITR yields a better recall than INTHELEX for all users except one (user 23).

For pairwise comparison of the two methods, the nonparametric Wilcoxon signed rank test was used [5], since the number of independent trials (i.e., users) is relatively low and does not justify the application of a parametric test, such as the t-test. In this experiment, the test was adopted in order to evaluate the difference in effectiveness of the profiles induced by the two systems according to the metrics pointed out in Table 3. Requiring a significance level $p < 0.05$, the test revealed that there is a statistically significant difference in performance both for Precision (in favor of INTHELEX) and for Recall (in favor of ITR), but not as regards Accuracy.

Going into more detail, as already stated, ITR performed very poorly only on user 23, whose interests turned out to be very complex to be captured by the probabilistic approach. Actually, all but one rates given by such a user were positive (ranging between 6 and 8), that could be the reason for such a behaviour. With respect to the complete dataset of all users, the accuracy calculated on the subset of all users except user 23 becomes statistically significant in favor of ITR.

```
likes(A) :-
  learn(A),
  mach(A),
  intellig(A),
  slot_title(A, F),
  slot_authors(A, G),
  slot_annotation(A, B),
  intellig(B, C),
  learn(B, D),
  occ_12(D),
  mach(B, E),
  OCC_12(E).
```

Fig. 3. Rule learned by INTHELEX

Table 4 reports the results about training and classification time of both systems. Training times vary substantially across the two methods. ITR takes an average of 6,2414 msec to train a classifier for a user when averaged over all 10 users. Training INTHELEX takes more time than ITR, but this is not a real problem because profiles can be learnt by batch processes without noise for users. In user profiling application, it is important to quickly classify new instances, for example to provide users with on-line recommendations. Both methods are very fast in this regard.

In summary, the probabilistic approach seems to have better recall, thus showing a trend to classify unseen instances as positive; on the contrary, the first-order approach tends to adopt a more cautious behavior, and classify new instances as negative. Such a difference is probably due to the approach adopted: learning in INTHELEX is data-driven, thus it works bottom-up and keeps in the induced definitions as much information as possible from the examples. This way, requirements for new observations in order to be classified as positive are more demanding, and few of them pass; on the other hand, this ensures that those that fulfill the condition are actually positive instances.

Another remark worth noting is that theories learned by the symbolic system are very interesting from a human understandability viewpoint, in order to be able to explain and justify the recommendations provided by the system. Figure 3 shows one such rule, to be interpreted as "the user likes a book if its annotation contains stems *intellig*, *learn* (1 or 2 times) and *mach* (1 or 2 times)". Anybody can easily understand that this user is interested in books concerning artificial intelligence and, specifically, machine learning.

In a commercial Web site perspective, the probabilistic behavior should be preferable. It could be used in developing recommender systems exploiting the *ranked list* approach for presenting items to the users. In this scheme, users specifies their needs in a form and the system presents a usually long list of results, ordered by their predicted relevance. On the other hand, the ILP approach could be adopted in situations when the system transparency is a critical factor and it is important to provide an explanation of why a recommendation was made.

From what said above, it seems that the two approaches compared in this paper have complementary *pros and cons*, not only as regards the representation language, but also as concerns the predictive performances. This naturally leads to think that some cooperation could take place between the two in order to reach higher effectiveness of the recommendations. For instance, since the probabilistic theories have a better recall, they could be used for selecting which items are to be presented to the user. Then, some kind of filtering could be applied on them, in order to present to the user first those items that are considered positive by the symbolic theories, that are characterized by a better precision.

5 Conclusions

Research presented in this paper has focused on methods for learning user profiles which are predictively accurate and comprehensible. Specifically, an intensive comparison between an ILP and a probabilistic approach to learning models of users' preferences was carried out. Experimental results highlight the usefulness and drawbacks of each one, that can suggest possible ways of combining the two approaches in order to offer better support to users accessing e-commerce virtual shops or other information sources. In particular, we suggest a simple possible way of obtaining a cascade hybrid method. In this technique, the probabilistic approach could be employed first to produce a coarse ranking of candidates and

the ILP approach could be used to refine the raccomandations from among the candidate set.

Currently we are working on the integration in INTHELEX of techniques able to manage numeric values, in order to treat in a more efficient way numerical features of instances, and hence to abtain theories with a more fine grean size.

References

[1] M. Degemmis, P. Lops, G. Semeraro, and F. Abbattista. Extraction of user profiles by discovering preferences through machine learning. In M. A. Klopotek, S. T. Wierzhon, and K. Trojanowski, editors, *Information Systems: New Trends in Intelligent Information Processing and Web Mining*, Advances in Soft Computing, pages 69–78. Springer, 2003.

[2] F. Esposito, G. Semeraro, N. Fanizzi, and S. Ferilli. Multistrategy Theory Revision: Induction and abduction in INTHELEX. *Machine Learning*, 38(1/2):133–156, 2000.

[3] D. Mladenic. Text-learning and related intelligent agents: a survey. *IEEE Intelligent Systems*, 14(4):44–54, 1999.

[4] Raymond J. Mooney and Loriene Roy. Content-based book recommending using learning for text categorization. In *Proceedings of the 5^{th} ACM Conference on Digital Libraries*, pages 195–204, San Antonio, US, 2000. ACM Press, New York, US.

[5] M. Orkin and R. Drogin. *Vital Statistics*. McGraw-Hill, New York, 1990.

[6] M. Pazzani and D. Billsus. Learning and revising user profiles: The identification of interesting web sites. *Machine Learning*, 27(3):313–331, 1997.

[7] G. Salton and M.J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, New York, 1983.

[8] G. Semeraro, F. Esposito, D. Malerba, N. Fanizzi, and S. Ferilli. A logic framework for the incremental inductive synthesis of datalog theories. In M. A. Klopotek, S. T. Wierzhon, and K. Trojanowski, editors, *Logic Program Synthesis and Transformation*, number 1463 in Lecture Notes in Computer Science, pages 300–321. N. E. Fuchs, 1998.

Monitoring the Evolution of Web Usage Patterns

Steffan Baron[1] and Myra Spiliopoulou[2]

[1] Institute of Information Systems, Humboldt-Universität zu Berlin
Spandauer Str. 1, Berlin 10178, Germany
sbaron@wiwi.hu-berlin.de
[2] Institute of Technical and Business Information Systems,
Otto-von-Guericke-Universität Magdeburg,
PO Box 4120, Magdeburg 39016, Germany
myra@iti.cs.uni-magdeburg.de

Abstract With the ongoing shift from off-line to on-line business processes, the Web has become an important business platform, and for most companies it is crucial to have an on-line presence which can be used to gather information about their products and/or services. However, in many cases there is a difference between the intended and the effective usage of a web site and, presently, many web site operators analyze the usage of their sites to improve their usability. But especially in the context of the Internet, content and structure change rather quickly, and the way a web site is used may change often, either due to changing information needs of its visitors, or due to an evolving user group. Therefore, the discovered usage patterns need to be updated continuously to always reflect the current state.
In this article, we introduce PAM, an automated *Pattern Monitor*, which can be used to observe changes to the behavior of a web sites visitors. It is based on a temporal representation of rules in which both the content of the rule and its statistical properties are modeled. It observes pattern change as evolution of the statistical measurements captured for a rule throughout its entire lifetime and notifies the user about interesting changes within the rule base. We present PAM in a case study on the evolution of web usage patterns. In particular, we discovered association rules from a web-servers log that show which pages tend to be visited within the same user session. These patterns have been imported into the monitor, and their evolution throughout a period of 8 months has been analyzed. Our results show that PAM is particularly suitable to gain insights into the changes a rule base is affected of over time.

1 Introduction and Related Work

Knowledge discovery is an iterative process that reflects the need of extracting knowledge from data that accumulate constantly [7]. The application expert periodically invokes a data mining tool to extract patterns from the data. Each invocation contributes new insights on the application domain, enriching the expert's domain knowledge and, occasionally,

motivating her to revise her beliefs. As the expert becomes gradually familiar with the patterns being extracted, she is increasingly interested in *changes* rather than in already known patterns. Especially in the context of the Internet, content and structure change rather quickly, and the way a web site is used may change often or even permanently, either due to the changing information needs of its visitors, or due to an evolving user group. Therefore, the discovered usage patterns need to be updated continuously to always reflect the current state. In the case of a heavily used web site with thousands of users per day, the question arises how this can be achieved with a reasonable effort.

One major problem is the large number of discovered rules, and the identification of *interesting patterns* has become a widely discussed topic [10,8]. Even for relatively small datasets this a problem and makes the inspection of all rules impractical. In some cases, it may be possible to assess the interestingness of rules manually or with respect to the application context. However, there are a number of approaches that offer potential solutions to this problem in an application independent manner [22,13,14].

When data, as in the case of web-server logs, is continuously collected over a potentially long period, the concepts reflected in the data will change over time. Due to internal and/or external factors, the distribution and/or the structure of the dataset may change. This requires the user to monitor the discovered patterns continuously, which is of particular importance for applications that timestamp data. One possible way to deal with the temporal dimension is to use an appropriate partitioning scheme. However, if the partitions are too big or too small it may be the case that important rules and/or changes are missed. Normally, partitioning is highly application depended. However, there is research into formal methods for application independent partitioning of data. Chen and Petrounias focus on the identification of valid time intervals for previously discovered association rules [10]. They propose a mechanism that finds (a) all contiguous time intervals during which a specific association holds, and (b) all interesting periodicities that a specific association has. Chakrabarti et al. propose the discovery of surprising, i.e. unexpected and therefore interesting, patterns in market basket analysis by observing the variation of the correlation of the purchases of items over time [9]. The underpinnings of that work come from time series analysis, so that the emphasis is on partitioning the time axis into such intervals that the rule statistics change dramatically between two consecutive intervals.

In the last years, a number of methods and techniques for maintaining and updating previously discovered knowledge have emerged which are able to deal with dynamic datasets. A widely used approach is that of *incremental mining* in which the knowledge about already extracted patterns is re-used in subsequent periods. Originally, the emphasis of incremental mining was on optimizing the miner's performance from one invocation of the miner to the next. Most of this research focuses on the updation of association rules [11,2,21,23], frequent sequences [24], and clusters [12]. They aim at efficiently updating the content of discovered rules, thus avoiding a complete re-run of the mining algorithm on the entire updated dataset.

The DELI Change Detector of Lee et al. uses a sampling technique to detect changes that may affect previously discovered association rules [18,17]. It invokes an incremental miner to modify the patterns if this turns out to be necessary.

Ganti et al. propose the DEMON framework for data evolution and monitoring across the temporal dimension [16]. DEMON focuses on detecting systematic vs. non-systematic changes in the data and on identifying the data blocks (along the time dimension), which have to be processed by the data miner in order to extract new patterns. The emphasis is on actualizing the knowledge base by detecting changes in the data, rather than detecting changes in the patterns.

Another avenue of research concentrates on the similarity of rules and on the statistical properties of rules by considering the *lifetime* of patterns, i.e., the time in which they are sufficiently supported by the data [15,9,10,16]. Ganti et al. propose the framework FOCUS for the comparison of two datasets and the computation of an interpretable, qualifiable deviation measure between them, whereby the difference is expressed in terms of the model the datasets induce [15].

However, all of these proposals consider only part of a pattern, either its content, i.e., the relationship in the data the pattern reflects, or the statistical properties of the pattern. In [3], we took a first step towards an *integrated* treatment of these two aspects of a rule. We proposed the *Generic Rule Model* (GRM) which models both the content and the statistics of a rule as a temporal object. Based on these two components of a rule, different types of pattern evolution were defined. Additionally, a simple monitor was implemented which used a user supplied deviation threshold to identify interesting changes to pattern statistics.

Pattern change is usually caused by concept drift. As Kelly et al. point out in [20], adaptive classification algorithms, such as adaptive Bayesian

networks are designed to overcome concept drift by considering the impact of each individual new record on the existing classifier and adapting it accordingly. However, the rapid accumulation of records, as in web-server logs, makes the consideration of the impact of each record ineffective. Moreover, individual records that come in large numbers are noisy and reflect trends only partially, especially for trends that manifest themselves slowly, such as a change in preferences or demographics of a user group.

The problem of interestingness arises also when evaluating the changes that have affected a rule. A commonly used approach to protect the domain expert from inspecting too many rule changes is the definition of limits for e.g. the steepness of change for the observed statistical measurements. When these limits are exceeded, the corresponding rule change is considered to be interesting.

The temporal aspects of patterns are taken into account in the rule monitors of [1,8] and [3,4,6]. In [8], Liu et al. count the significant rule changes across the temporal axis. They pay particular attention on rules that are "stable" over the whole time period, i.e. do not exhibit significant changes, and juxtapose them with rules that show trends of significant increase or decrease. Significance tests form the basis of the experiments. In [1], upward and downward trends in the statistics of rules are identified using an SQL-like query mechanism. Closer to our work is the research of Liu et al. on the discovery of "fundamental rule changes" [19]: they consider rules of the form $r_1, \ldots, r_{m-1} \rightarrow r_m$, and detect changes on support or confidence between two consecutive timepoints by applying a χ^2-test. In our previous work [3,4], we model rules as temporal objects, which may exhibit changes of statistics *or content* during the observation period, and we focus on surprising changes, such as the disappearance of a rule and the correlated changes of pairs of rules. In [6], we make the distinction between "permanent" rules that are always present (though they may undergo significant changes) and those that appear only temporarily and indicate periodic trends, and discuss methods for identifying them in a *progressive study.*

In this study, we present PAM, an automated pattern monitor, and its theoretical underpinnings. In a case study on the evolution of web usage patterns, we show how the mechanisms implemented by PAM can be used to identify interesting changes in the usage behavior. In particular, we discovered association rules from a web-servers log that show which pages tend to be visited within the same user session. These patterns have been imported into the monitor, and their evolution throughout a period of 8 months has been analyzed.

In the following section, we will introduce the theoretical framework PAM is based upon and its architecture. In Sec. 3 our experimental results are summarized. Sec. 4 concludes our study.

2 Theoretical Framework

In [5], we introduce the theoretical foundations of PAM which are briefly described in the following. PAM builds on a temporal rule model which consists of a rules' content, i.e., the relationship in the data the pattern reflects, and the statistical measurements captured for the rule.

2.1 Temporal Rule Model

As basis for the temporal representation of patterns the *Generic Rule Model* (GRM) is used [3]. According to the GRM, a rule R is a temporal object with the following signature:

$$R = ((ID, query, body, head), \{(timestamp, statistics)\})$$

In this signature, ID is a system generated identifier, which ensures that all rules with the same body (antecedent) and head (consequent) have the same ID. It is used to identify a rule non-ambiguously throughout its entire lifetime. The *query* is the data mining query or similar specification of values for the mining parameters. Note that *query* and ID are invariant across the time axis. Contrary to it, the statistics may vary between two timestamps. The statistics depend on the rule type: We currently consider the support, confidence and certainty factor of association rules. A detailed discussion of the components of the rule signature can be found in [3].

2.2 Detecting Significant Pattern Changes

A mechanism that identifies changes to a rules statistic which exhibit a particular strength we denote as *change detector*. We use the notion of statistical significance to assess the strength of pattern changes. In particular, we use a two-sided binomial test to verify whether an observed change is statistically significant or not.

For a pattern ξ and a statistical measure s at a time point t_i it is tested whether $\xi.s(t_{i-1}) = \xi.s(t_i)$ at a confidence level α. The test is applied upon the subset of data D_i accumulated between t_{i-1} and t_i, so that the null hypothesis means that D_{i-1} is drawn from the same population as

D_i, where D_{i-1} and D_i have an empty intersection by definition. Then, for a pattern ξ an alert is raised for each time point t_i at which the null hypothesis is rejected.

Example. Let $sup(t_{i-1}) = 0.1825$ be the support of pattern ξ at time point t_{i-1}, i.e. over dataset D_{i-1}. Let the number of sessions in D_i that support the pattern be 608 (*successes*) upon a total of 2914 sessions in D_i. We test the null hypothesis H_0 that the support of ξ has not changed significantly:

$$H_0: \quad \xi.sup_{i-1} = \xi.sup_i$$
$$H_1: \quad \xi.sup_{i-1} \neq \xi.sup_i$$

At $\alpha = 0.01$, the confidence interval ranges from 0.1896 to 0.2287. Since the true support value at time point t_{i-1} is smaller than the lower boundary of the confidence interval we reject H_0 and state that the support has changed significantly from time point t_{i-1} to time point t_i.

These tests are applied to the set of all patterns that appear in a given period. All significant pattern changes are additionally checked for their temporal dimension, i.e., whether they are only of temporary nature. We differentiate between two cases, (a) the value of the statistic returns immediately to its previous level, and (b) the value remains stable at the new level for at least one more period. For this purpose, we again use the binomial test and check if there is a significant change in period t_{i+1} annulling the change in period t_i. If so, the second change is not reported to the user because it only represents the return of the measure to its actual level. Instead, the significant pattern change is marked as a *core alert* which may be used as an indicator for a beginning concept drift.

2.3 Heuristics for Detecting Interesting Pattern Changes

As opposed to the change detector, the heuristics are used to track changes to the statistics of patterns starting at the time point at which they have emerged for the first time, even if they are not continuously in the rule base. Therefore, they can reveal potentially interesting changes also for those patterns that do not satisfy the thresholds given in the mining query in a particular period. Since the change detector is only aware of patterns that are present in the rule base, this property may be of particular interest to the analyst.

Occurrence-based Grouping. Patterns observed in a particular period reflect the properties of the underlying dataset within this specific

time interval. On the other hand, patterns that are present in each period reflect (part of) the invariant properties of the population. If such patterns change this may be of particular interest to the user.

We, therefore, group rules with respect to their *stability* over time and use the term *occurrence* to denote the proportion of periods in which the rule is present, i.e., the percentage of time points in which its statistics exceeded the threshold values specified in the mining query. In particular, let f be the frequency of appearance of a pattern, defined as the ratio of time points at which the observed statistic measure exceeds the threshold specified by the domain expert. The range of f is $[0, 1]$, which we partition into the intervals $I_L = [0, 0.5)$, $I_M := [0.5, 0.75)$, $I_H := [0.75, 0.9)$, $I_{H+} := [0.9, 1)$ and $I_{permanent} := [1, 1]$. Alternatively, $I_{permanent}$ can be set to $[0.9, 1]$ for large values of n, i.e. for a large number of discrete time points. Then, we label a pattern ξ as L, M, H, $H+$ or *permanent* according to the interval at which $f(\xi)$ belongs. Patterns labeled as H or $H+$ are characterized as *frequent* patterns, whereas L and M form the set of *temporary* patterns.

Then, for a pattern ξ an alert is raised for each time point t_i at which the pattern changes the group it belongs to. Certainly, in order to assess the reliability of patterns, a sufficiently long training phase has to be passed through before meaningful group changes can be observed; in the short run, this approach will be very sensitive to patterns that vanish or emerge.

As already mentioned above, one important peculiarity of this approach is that it can identify many significant rule changes which cannot be observed by significance tests. This is caused by the fact that this method will raise an alert when a rule appears/vanishes in/from the rule base. In such cases, changes to the statistics of a rule will usually be significant. However, if a pattern disappears from the rule base its time series is interrupted and a significance test cannot be applied.[1]

Corridor-based Heuristic. For this heuristic, we define a *corridor* around the time series of a pattern. A corridor is an interval of values which is dynamically adjusted at each time point to reflect the range of values encountered so far. In particular, for a pattern ξ and a statistic measure s, we compute the mean m_i and standard deviation $stddev_i$ of the values $\{\xi.s(t_j)|j = 0, \ldots, i\}$. The corridor at time point t_i is defined as the interval $I(t_i) := [m_i - stddev_i, m_i + stddev_i]$, having a width of one

[1] One possible way to bypass this problem is to use the pattern monitor proposed in [6] which computes the statistics of a rule directly from the underlying dataset.

standard deviation in each direction of the mean. Then, for pattern ξ an alert is raised for each time point t_i at which the value of the time series is outside the corridor $I(t_i)$.

The corridor-based heuristic takes account of the values already encountered for a given pattern. It is insensitive to oscillations of the time series around the threshold value used to discover rules. However, it is sensitive to changes that depart from past values but still remain in the interval, and it can only be defined reasonably for late time points: at time point t_1, a pattern change is most likely to be signaled because the mean and the standard deviation are not well-defined. Thus, the corridor-based heuristic is more appropriate for a retrospective study of the data; for a progressive study, a sufficient number of mining sessions must be performed first. As in the case of the occurrence-based grouping of patterns, this approach may also identify significant changes which cannot be covered by significance tests because corridor violations are tracked starting in the first period a pattern is visible. Therefore, an alert may also be raised when the pattern has disappeared from the rule base. However, such changes may be important for the user, e.g. if the reason for the disappearance is of interest.

Interval-Based Heuristic. For this heuristic, we partition the range of values of the time series into *intervals* of equal width. In particular, we consider the interval $[\tau, M]$, where M is the maximum permissible value per definition of the statistical measure under observation (e.g. support), while τ can be either a threshold provided by the application expert or the minimum permissible value per definition of the statistical measure. This range is partitioned into k equal subintervals. Then, for pattern ξ an alert is raised for each time point t_i at which the value of the time series is in a different interval than for t_{i-1}.

Example. Consider a time series on rule support and a pattern ξ, whose time series on support we denote as $\xi.sup(t_i), i = 0, \ldots, n$. Further, assume that $k = 4$, i.e. the range should be split into four intervals. Then, the range is $[\tau_{sup}, 1]$, where τ_{sup} is the support threshold specified in association rules' discovery.[2] For $\tau_{sup} = 0.2$, we would have four intervals, namely $I_1 = [0.2, 0.4)$, $I_2 = [0.4, 0.6)$, $I_3 = [0.6, 0.8)$ and $I_4 = [0.8, 1]$. A pattern change is signaled for ξ at each $t_i, i \geq 1$ such that

$$\xi.sup(t_i) \in I_j \wedge \xi.sup(t_{i-1}) \in I_{j'} \wedge I_j \neq I_{j'}.$$

[2] This threshold is part of the *query* in the signature of a rule.

2.4 Identifying Atomic Changes

The change detector returns at each time point t_i the set of all patterns, whose observed statistic measure has changed with respect to the previous period. Normally, this set will be large, and some of the patterns may be correlated because they overlap in content. In such cases, it is likely that their changes are due to the same drift in the population. Therefore, we try to identify a minimal set of patterns that caused all change. For this purpose, we consider the components of each pattern, assuming that if a pattern change has occurred, it may be traced back to changes of the statistics of its components. We use the term *atomic change* for a change in a pattern which has no components that has itself experienced a change.

According to a rule's signature in Sec. 2.1, a rule has a body and a head. We observe them together as components of a pattern ξ that corresponds to the rule's itemset. If a pattern change on ξ is reported at time point t_i, it may be due to a change of one or more of its components. Therefore, at time point t_i we consider all combinations of the elements $e_1, \ldots, e_{length(\xi)}$ constituting ξ, i.e. $c(\xi) := \sum_{j=1}^{length(\xi)-1} \binom{length(\xi)}{j}$ components, excluding ξ itself. The following algorithm is used to attribute support changes of a rule to the support changes of its components:

> Let d be the detector used to raise alerts for changes in patterns.
> Let Ξ be the collection of patterns, for which an alert has been raised by d.
> We order the patterns in Ξ by length, keeping longer patterns first. Then, for each $\xi \in \Xi$
>
> 1. For each component ζ of ξ with length $length(\xi) - 1$:
> 2. If ζ is in Ξ
>   then we mark ξ as "removed" and break the loop
>   else we continue with the next component.

Undeleted patterns in Ξ are presented to the application expert. The algorithm assumes the base characteristic of association rules, namely that if a rule is frequent, then all its components are also frequent. Hence, for each pattern in the rule base produced at each time point t_i, all its sub patterns are also in the rule base. Since the detector considers all time series, any pattern that has experienced a change at t_i is placed in Ξ, independently of its components. Thus, if a component of a pattern ξ is found in Ξ, we attribute the change of ξ to it and ignore ξ thereafter. Since the algorithm moves progressively towards smaller patterns, a component

that was found to be responsible for a change in a pattern may itself be removed.

2.5 Architecture of PAM

Technically spoken, PAM encapsulates one or more data mining algorithms and a database back-end that stores data and mining results. The general structure of a PAM instance is depicted in Fig. 1. The core of

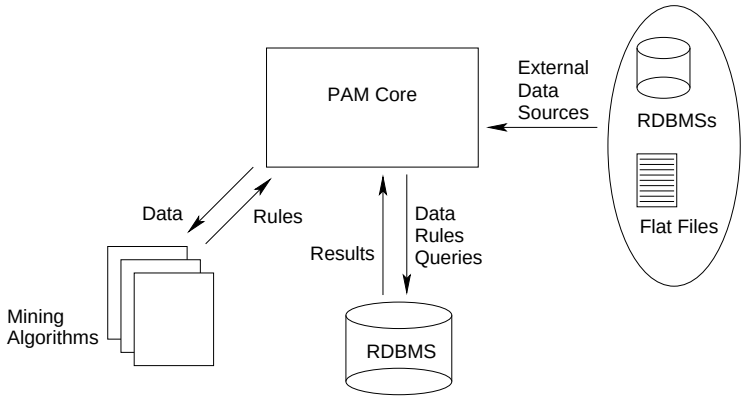


Figure 1. General structure of a PAM instance.

PAM implements the change detectors and heuristics described in the previous paragraphs. When incorporating new data, e.g. the transactions of a new period, these algorithms are used to identify interesting pattern changes. The core offers interfaces to external data sources like databases or flat files. In the database back-end not only the rules discovered by the different mining algorithms are stored, but also the (transaction) data. For example, in order to conduct significance tests it is sometimes necessary to access the base data. Mining results are stored according to the GRM which has been transformed into a relational schema. At present, any mining algorithm implemented in the Java programming language can be used within PAM.

3 Experiments

For the experiments, we used the log file of a web-server hosting a non-commercial web site, spanning a period of 8 months in total. All pages

on the server have been mapped to a concept hierarchy reflecting the purpose of the respective page. The sessionized log file has been splitted on a monthly basis, and association rules showing the different concepts accessed within the same user session have been discovered. In the mining step, we applied the association rule miner of the WEKA tools [25] using minimum support of 2.5% and minimum confidence of 80%.

Tab. 1 gives a general overview on the evolution of the number of page accesses, sessions, frequent single items and rules found in the respective periods. The last three columns give the number of rules that have emerged, remained in the rule base, or disappeared from one period to the next. The first period corresponds to the month October. It can

period	# accesses	# sessions	# freq. items	# rules	# new	# old	# disapp.
1	8335	2547	20	22	22	–	–
2	9012	2600	20	39	27	12	10
3	6008	1799	20	26	13	13	26
4	4188	1222	21	24	12	12	14
5	9488	2914	20	14	1	13	11
6	8927	2736	20	15	6	9	5
7	7401	2282	20	13	5	8	7
8	9210	3014	20	11	3	8	5

Table 1. General overview on the dataset.

be seen that in periods 3 and 4 (December and January) the site was visited less frequently than in other periods, although the number of discovered rules remains comparably high. The number of frequent single items is rather invariant, whereas the number of rules falls, especially in the second half of the analysis.

In order to assess the stability of the rules found, we grouped them according to their occurrence, i.e., the share of periods in which they were present (Tab. 2). In total, there were 58 distinct rules but only 11 rules are frequent according to the definition in Sec. 2.3, i.e. were present in at least 6 periods. Due to the large proportion of rules which appear only once or twice, there are strong changes to the mining results even for adjacent periods, especially in the first half of the analysis (cf. Tab. 1).

3.1 Applying Significance Tests and Heuristics

All experiments are solely based on the support of rules, i.e., in order to determine and assess rule changes only the support of a pattern was

# periods present	# rules	share
8	3	100.0
7	4	87.5
6	4	75.0
5	2	62.5
4	2	50.0
3	6	37.5
2	15	25.0
1	22	12.5

Table 2. Rules grouped by occurrence.

analyzed. However, when analyzing changes to the confidence of a rule the same methodology can be used. As pointed out in Sec. 2.2, we differentiate between short and long-term changes to the statistics of rules. For simplicity, we considered only two different scenarios in the experiments, either the value returns immediately to its previous level, or it remains the same for at least one more period (core alert).

The significance tests were applied to all cases throughout the entire case study.[3] In total, we observed 164 cases in 8 periods, from which 142 cases were checked for significance.[4] 17 support changes turned out to be also significant, from which 4 changes were core alerts. Tab. 3 shows the results of the significance tests and the results of the occurrence-based grouping and the corridor-based approach.[5] In the second column

approach	# changes	# significant	# core alerts
change detector	142	17	4
occurrence-based grouping	49	12	3
corridor-based heuristic	107	32	6

Table 3. Results of the different approaches.

the number of raised alerts is given. Obviously, the number of alerts is even greater than the number of cases to which the change detector was

[3] The term *case* refers to the appearance of one rule in one period.
[4] Cases from the first period cannot be checked for significance.
[5] For the interval-based heuristic we determined only one interesting change. It turned out that the support values of the rules were too small and that a large number of intervals would have been necessary to get useful results. We therefore decided to skip this approach.

applied. We noticed that the intersection of changes identified by one of the heuristics and the total number of cases throughout the analysis amounted only to 33. This complies with our assumption from Sec. 2.3 that the heuristics also reveal interesting changes for patterns that are invisible at the moment. Therefore, the change detector should be used in conjunction with at least one of the heuristics.

As described in Sec. 2.3, the heuristics may be used only after a sufficiently long training period. Due to the limited number of periods, we used the first 4 periods as a training phase and observed group changes only for the remaining periods. Fig. 2 shows examples of applying the occurrence-based grouping of patterns and a time series of the corridor-based heuristic. In the left figure, the horizontal lines at 0.5, 0.75 and

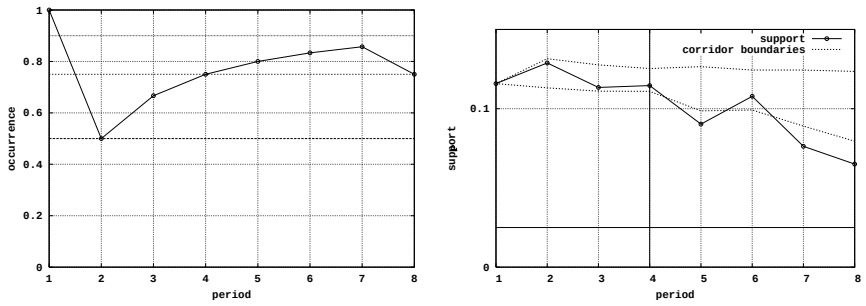


Figure 2. Examples of occurrences and corridors.

0.9 represent the borders of the different groups of relative occurrence. In the first period the pattern is present (occurrence = 1), in the second period the pattern disappears and the occurrence drops to 0.5. However, since the training phase is not yet finished we do not observe a group change. In the remaining periods, the pattern is present and the occurrence grows, except for period 8 in which the pattern again disappears. In the other figure, it is shown how the corridor reacts on the changing support. In periods 5 and 7 we observe corridor violations. Although the support value for the last period is also outside the corridor no alert is raised. Instead, the alert in period 7 is marked as a core alert which may signal a beginning concept drift.

In summary, there were basically two different results: on the one hand, we had a small number of permanent patterns which changed only slightly throughout the analysis. On the other hand, there were many

temporary patterns, especially in the first half of the analysis, which changed almost permanently. For example, it turned out that rules which were present in only 2 periods were responsible for 31 corridor violations, whereas the permanent patterns caused only 8 violations.

3.2 Detecting Atomic Changes

For the results of the change detector and the heuristics we decomposed the rules and checked their itemsets for significant changes. Tab. 4 summarizes the results of this analysis. Again, the number of (significant)

approach	# changes	# significant
change detector	59	28
occurrence-based grouping	50	21
corridor-based heuristic	143	73
corridor & interval	156	77

Table 4. Number of significant itemset changes.

changes is much larger for the heuristics. Interestingly, in this case it seems sufficient to use only the corridor approach since it reveals almost all significant itemset changes. It can be seen that, no matter which approach is considered, only about 50% of the itemsets in a pattern that changed significantly also show significant changes.

4 Conclusions

In this article, we introduced PAM, an automated pattern monitor, which was used to observe changes in web usage patterns. PAM is based on a temporal rule model which consists of both the content of a pattern and the statistical measures captured for the pattern. Moreover, we described heuristics to identify not only significant but also interesting rule changes which take different aspects of pattern reliability into account. We argue that concept drift as the initiator of pattern change often manifests itself gradually over a long period of time where each of the changes may not be significant at all. Additionally, if a pattern disappears from the rule base it is important to know why it disappeared. We presented PAM in a case study on web usage mining, in which association rules were discovered that show which types of web pages tend to be viewed in the same user

session. Our results show that PAM reveals interesting insights into the evolution of the usage patterns

Challenging directions for future work include the application of the methodology to the specific needs of streaming data, and the identification of interdependencies between rules from different data partitions. In this scenario, each partition constitutes a transaction in each of which a number of items (rules) appear. These *transactions* can then be analyzed to discover periodicities of pattern occurrence and/or relationships between specific rules.

Acknowledgements: Many thanks to Gerrit Riessen for his helpful advice.

References

1. R. Agrawal, G. Psaila, E. L. Wimmers, and M. Zaït. Querying shapes of histories. In *VLDB*, Zürich, Switzerland, 1995.
2. N. F. Ayan, A. U. Tansel, and E. Arkun. An Efficient Algorithm To Update Large Itemsets With Early Pruning. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 287–291, San Diego, CA, USA, August 1999. ACM.
3. S. Baron and M. Spiliopoulou. Monitoring change in mining results. In *3rd Int. Conf. on Data Warehousing and Knowledge Discovery - DaWaK 2001*, Munich, Germany, Sept. 2001.
4. S. Baron and M. Spiliopoulou. Monitoring the Results of the KDD Process: An Overview of Pattern Evolution. In J. M. Meij, editor, *Dealing with the Data Flood: Mining data, text and+multimedia*, chapter 6, pages 845–863. STT Netherlands Study Center for Technology Trends, The Hague, Netherlands, Apr. 2002.
5. S. Baron and M. Spiliopoulou. Detecting Long-Term Changes of Patterns over Accumulating Datasets. Submitted to *IEEE International Conference on Data Mining (ICDM'03)*, 2003.
6. S. Baron, M. Spiliopoulou, and O. Günther. Efficient Monitoring of Patterns in Data Mining Environments. In *Seventh East-European Conference on Advance in Databases and Information Systems (ADBIS'03)*, Dresden, Germany, Sep. 2003. Springer. To appear.
7. M. J. Berry and G. Linoff. *Data Mining Techniques: For Marketing, Sales and Customer Support*. John Wiley & Sons, Inc., 1997.
8. Y. M. Bing Liu and R. Lee. Analyzing the interestingness of association rules from the temporal dimension. In *IEEE International Conference on Data Mining (ICDM-2001)*, pages 377–384, Silicon Valley, USA, November 2001.
9. S. Chakrabarti, S. Sarawagi, and B. Dom. Mining surprising patterns using temporal description length. In A. Gupta, O. Shmueli, and J. Widom, editors, *VLDB'98*, pages 606–617, New York City, NY, Aug. 1998. Morgan Kaufmann.
10. X. Chen and I. Petrounias. Mining Temporal Features in Association Rules. In *Proceedings of the 3rd European Conference on Principles of Data Mining and Knowledge Discovery*, Lecture Notes in Computer Science, pages 295–300, Prague, Czech Republic, September 1999. Springer.

11. D. W. Cheung, S. Lee, and B. Kao. A general incremental technique for maintaining discovered association rules. In *DASFAA'97*, Melbourne, Australia, Apr. 1997.
12. M. Ester, H.-P. Kriegel, J. Sander, M. Wimmer, and X. Xu. Incremental Clustering for Mining in a Data Warehousing Environment. In *Proceedings of the 24th International Conference on Very Large Data Bases*, pages 323–333, New York City, New York, USA, August 1998. Morgan Kaufmann.
13. A. Freitas. On objective measures of rule surprisingness. In *PKDD'98*, number 1510 in LNAI, pages 1–9, Nantes, France, Sep. 1998. Springer-Verlag.
14. P. Gago and C. Bento. A Metric for Selection of the Most Promising Rules. In J. M. Zytkow and M. Quafafou, editors, *Principles of Data Mining and Knowledge Discovery, Proceedings of the Second European Symposium, PKDD'98*, Nantes, France, 1998. Springer.
15. V. Ganti, J. Gehrke, and R. Ramakrishnan. A Framework for Measuring Changes in Data Characteristics. In *Proceedings of the Eighteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pages 126–137, Philadelphia, Pennsylvania, May 1999. ACM Press.
16. V. Ganti, J. Gehrke, and R. Ramakrishnan. DEMON: Mining and Monitoring Evolving Data. In *Proceedings of the 15th International Conference on Data Engineering*, pages 439–448, San Diego, California, USA, February 2000. IEEE Computer Society.
17. S. Lee, D. Cheung, and B. Kao. Is Sampling Useful in Data Mining? A Case in the Maintenance of Discovered Association Rules. *Data Mining and Knowledge Discovery*, 2(3):233–262, September 1998.
18. S. D. Lee and D. W.-L. Cheung. Maintenance of Discovered Association Rules: When to update? In *ACM-SIGMOD Workshop on Data Mining and Knowledge Discovery (DMKD-97)*, Tucson, Arizona, May 1997.
19. B. Liu, W. Hsu, and Y. Ma. Discovering the set of fundamental rule changes. In *7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2001)*, pages 335–340, San Francisco, USA, August 2001.
20. N. M. A. Mark G. Kelly, David J. Hand. The Impact of Changing Populations on Classifier Performance. In *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 367–371, San Diego, Aug. 1999. ACM.
21. E. Omiecinski and A. Savasere. Efficient Mining of Association Rules in Large Databases. In *Proceedings of the British National Conference on Databases*, pages 49–63, 1998.
22. P.-N. Tan, V. Kumar, and J. Srivastava. Selecting the Right Interestingness Measure for Association Patterns. In *Proc. of the Eighth ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining*, 2002.
23. S. Thomas, S. Bodagala, K. Alsabti, and S. Ranka. An Efficient Algorithm for the Incremental Updation of Association Rules in Large Databases. In *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining (KDD-97)*, pages 263–266, Newport Beach, California, USA, Aug. 1997.
24. K. Wang. Discovering patterns from large and dynamic sequential data. *Intelligent Information Systems*, 9:8–33, 1997.
25. I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools with Java implementations*. Morgan Kaufmann, San Francisco, 1999.

Beyond user clicks: an algorithm and an agent-based architecture to discover user behavior

E. Menasalvas (1[†]), S. Millán (2), M. Pérez (1), E. Hochsztain (3[*]), V. Robles (1), O.Marbán (1), J. Peña (1), A. Tasistro (3)

(1) Facultad de Informática UPM. Madrid Spain (2) Universidad del Valle. Cali. Colombia.
(3) Facultad de Ingeniería - Universidad ORT Uruguay

Abstract. One of the gaps that has to be filled for the fully deployment of web activities is a closer human relationship with the user. This lack can only be dealt with understanding user online actions, which will make proactive and autonomous web behavior possible. In order to find out what navigators behavior looks like it is crucial to understand the semantics underlying their clicks. Being aware of current navigation patterns is not always enough. Knowledge about user objectives and session goals can be discovered from the information collected and tracked by web clicks. Nevertheless, information on these clicks has to be enriched with semantics if user behavior understanding is the ultimate goal. This paper presents an approach that provides an estimation of the end result of a user navigation in terms of site goals achievements. Once clickstream information has been enriched we propose to apply a method based on discriminant analysis in order to obtain two different results: (i) the relevant factors that contribute most to the success of a session and (ii) a statistical classification method to estimate the result of an ongoing session. In order to carry out this proposal, we also present the design of an agent-based architecture in which the role of each agent is deeply analyzed.

1. Introduction and related works

The Web is not only a technology to connect computers; it is also a new way of communication, cheaper and with greater location independency than the traditional one. This explains the amazing number of organizations that during the last decade have started their traditional activities in the Internet, designing and implementing web sites to interact with their customers.

It is possible to say that the web is becoming one of the main communication channels for any kind of transaction, being commerce obviously one of its main uses. However, web-based activities loose direct contact with clients and, therefore, fail to achieve many of the features that enable small businesses to develop a warm human relationship with customers. This must not be understood as if those features cannot

[+] Research is partially supported by Universidad Politécnica de Madrid (project Web-RT)
[*] Research has been partially supported by Programa de Desarrollo Tecnológico (Uruguay).

be translated onto the web. In this sense, it is important to say that many enterprises have been very worried about getting hold of the identity of the navigator. In the traditional way of making business, a good CRM means providing a user with the right product at the right moment. What is important is not the identity of the user but his or her likes and dislikes, his or her preferences, the way he or she behaves. Thus, trying to have a good e-CRM, the behavior of the user has to be analyzed and for doing so the only available data are the clickstream. In this sense, most of enterprises are investing great amounts of money establishing mechanisms to discover internet's user behavior.

Based on data mining techniques many approaches [9,16,21,23,25,29,31,34,35,38] have been proposed for tracking and analyzing clickstream data in order to obtain most frequent paths. Most of them calculate user profiles taking this information into account. Nonetheless, just knowing most frequent patterns is not enough; it is of vital importance to combine this information with company goals with the intention of making this process more profitable for web site sponsors.
Consequently, the challenge is to get, analyze and understand the behavior of every user that connects to the web site in order to improve information on the webhouse, make better decisions and take action. Based on annotations and ontologies, several methods and approaches have been proposed in order to take these site semantics into account [13, 26, 5, 17, 36, 19, 10, 33, 6]. A framework for web personalization that integrates domain ontologies and usage patterns is presented in [13]. Semantic information could be obtained from textual content included in the site or using conceptual hierarchies based on services [7]. An approach that takes into account associated information goals inferred from particular patterns and from information scent associated to linked pages is proposed in [11]. In [33] a site's pages are classified according to their function: action and target pages. On the other hand, in [26] user actions are represented based on an ontology's taxonomy. URL's are mapped to applications events depending on what they represent (actions or content).

In [20] we propose an algorithm that takes into account both the information of the server logs and the business goals improving traditional web analysis. In this algorithm, however, the value of the links is statically assigned. Although this approach is useful for finding the value of the path the user has taken, this value depends on the value of the links. In this paper, we propose a method based on discriminant analysis that makes it possible to establish factors that are relevant for determining the success or failure of a session with respect to business goals. Then based on these factors and applying a statistical classification method we provide the site administrator with a predictive model to estimate the result of an on-going session.
Considering that we can have dynamic pages, the algorithm does not take the page or link the user has visited as parameters. It takes instead certain concepts (e.g., factors, actions) extracted from them that are interesting for the site.
The key point in these approaches is how company goals are translated into parameters and values to be automatically managed during web server performance. These values are required to be able to predict the most valuable (in terms of the site goal) ongoing sessions. The different paths themselves are not the focus – what is

important is to discover, given a visited sequence of pages, the final result of the session and we propose to estimate this result based not on the pages the user has visited but on the semantics underlying these pages.
The point here to be emphasized is that the proposed approach is very effective as we are dealing with a predictive algorithm whose model though it has been calculated off-line by analyzing past sessions, needs to be online to be applied. The predictive model not only estimates the result of the session but also identifies the relevant factors that make a session successful. With these values, we provide the site administrator with enhanced information to find out which action should be performed next in order to provide the best service to navigators and to be more competitive. This information can be applied as an important parameter for web server load balancing, priority scheduling or even web server acceptance. Many other performance and tuning parameters can be affected by this goal-oriented analysis.

In spite of the huge volume of data stored in the web, the relationship between user navigation data and site effectiveness, in terms of site goals when trying to design "good pages" from the site users' point of view, is still difficult to understand. Several approaches, models and measures [8, 34, 3, 15, 22] have been proposed in order to evaluate and improve the success of web sites. Decision-making criteria related to design and content of web sites are needed so that user behavior matches the objectives and expectations of web site owners.

We propose to analyze how well a user's navigation fits company aims, how accurate web site content reflects company's purposes and how much web site structure contributes to achieve company's goals. In order to carry out this proposal, we introduce in this paper an architecture based on software agents. Taking into account that software agents exhibit a degree of autonomous behavior and attempt to act intelligently on behalf of the user for whom they are working [27, 25], web agents included in the architecture deal with all tasks of the proposed method.

In web domain, software agents have been used with several purposes: filtering, retrieval, recommending, categorizing and collaborating. Several systems based on filtering and searching agents have been proposed in order to assist site's users [12, 14, 30, 37].

Based on knowledge represented by multiple ontologies in [28] agents and services are defined in order to support navigation in a conference-schedule domain. ARCH (Adaptive Retrieval based on Concept Hierarchies) [32] helps the user in expressing an effective search query, using domain concept hierarchies.

Most of these systems have been designed as user-side agents to assist users to carry out different kinds of tasks. The agent-based architecture proposed in this paper has been designed taking into account the business point of view. Agents, in our approach could be considered as business-side agents.

The remainder of the paper is organized as follows. Section 2 introduces the way web logs have to be enriched in order to extract the semantics that we need. In section 3

the proposed methodology to calculate relevant factors and consequently estimate the result of the session is further explained. Section 4 describes the agent architecture defined for the deployment of the method analyzed in the previous section. Section 5 presents the experimental results obtained from applying the method. Section 6 presents the conclusions as well as suggestions for further research.

2. Preliminaries of the methodology

Due to different points of view, organizations can define different business goals. For example, the marketing department could be interested in the attractiveness and ease of use of the web site for the user and the department in charge of the design of the pages could be only interested in page design. In contrast, the sales department will be interested in the user buying some products. In order to capture this information, we assume that pages have been enriched with semantic information related both to the content (as already proposed in [6]) and to the business goals. In this paper we estimate, for each goal and for each viewpoint, the result of an ongoing session. In order to do so we assume that we already have information on past sessions and that these sessions have been already classified as success or failure sessions (a set of sessions is chosen as training set and classified by an expert). Then, based on this information we compute the end result of the session by means of a predictive algorithm that estimates the importance of visiting certain pages.
In fact, the algorithm will not only take into account the url of the page itself but the semantics underlying such a page. In order for this to be possible, information on pages has to be enriched with information both related to the semantics of the page, from the point of view of its content, and to the business goals.

Let $Goals = \{g_1, g_2, \ldots, g_s\}$ represent the goals of a company in a particular moment.
Let $Viewpoints = \{v_1, v_2, \ldots, v_n\}$ represent different point of views (e.g., marketing, sales).
Let w_{jj} be the function that assigns weights to each goal, where i represents the i^{th} viewpoint and j the j^{th} goal. We assume that weights are going to be assigned to different goals depending on the viewpoint.
As this function is a weight function, it has the following properties:

- $0 \leq w_{ij} = w(g_i, v_j) \leq 1$ where $g_i \in Goals$ and $v_j \in Viewpoints$
- $\sum_{i \in Goals} w_{ij} = 1$

Table 1 shows an example of possible weight assignment.

	Viewpoints		
Goals	Marketing	Sales	Design
Top of Mind	0.7	0,1	.05
Awards	0.1	0,05	0.9
High sales	0.2	0.85	0.05
Total	1	1	1

Table 1: Possible weight assignment

Let $\Omega = \{\alpha_1, ..., \alpha_m\}$ be the set of pages of a web site. This set includes all pages that can be dynamically generated.
We propose to assign semantics to each page. The approach has no problem with dynamic pages as the semantics is assigned to them when they are generated (it is not related to the url itself but to the content). According to Berendt [4], based on ontologies, an analyst can use several abstract criteria to order pages or groups of pages. In this sense, it is possible to have different ontologies associated to different points of view and business goals. We could also use content-based, service-based hierarchies [7] or domain knowledge contained in the site [13].
In our case, for each site and for each particular viewpoint we define the set of semantics (actions and/or contents) that are relevant to be studied by the site. Notice that this set of semantics can be modified along the site life depending on the things that are relevant to be analysed at each moment. According to this information, ontologies are defined.
Nevertheless, in this paper we are not concerned about how the ontology is created and maintained but about the way we use this information to obtain a predictive model that, using the semantics of the pages, can compute the value of an ongoing session. We will use a granular approach so that the value of a session can be estimated taking into account the semantics of each of the visited pages.

Once the logs have been enriched, sessions are classified as success or failure so that two tables, *Enriched session* and *Results,* are obtained. In order to do so, a training set of sessions is chosen and they are classified according to an expert, so that the result of the session is obtained. The same expert decides the relevant concepts (extracted from the ontologies) to enrich each session.
As a consequence, in the *enriched session* table, each tuple contains, apart from the session identifier, the concepts that are taken into account (i.e., information related to action, contents, design of the pages). For each piece of information taken into account, the session can only take values 0 or 1 to specify whether that action happened during the session or not. As we have already mentioned, this is a granular approach that takes into account relevant actions or behaviours at the site rather than in individual pages.
On the other hand, it is also necessary to obtain a *Results* table that contains information related to the success or failure of each session from each point of view considered. This is innovative in the sense that the proposed algorithm will not only extract relevant concepts for the site but relevant concepts of the site depending on the viewpoint considered. This way, for each session and for each goal we will have a measure of how successful the session happened to be. Once this information is available, the proposed analysis is performed for each viewpoint and for each goal.

3. Proposed Methodology: Predicting an ongoing session success or failure

For each pair (viewpoint, goal), a table is generated in which the attributes contain semantics extracted from the pages that can be visited during a session. The value of each attribute for each session will be 1 if the action took place in that session, or 0 otherwise. The last column of the table specifies, according to an expert, if the session was a success or a failure for a particular (viewpoint, goal) pair. This table is directly obtained from combining the *enriched session* with the *results* tables described above.
In a second step, we applied the stepwise multivariate predictive model to two sets of sessions designated as successful and failure n_1 and n_2 , respectively. The basic strategy in discriminant analysis is to define a linear combination of the dependent attributes in our case representing semantic actions or concepts $s_1,s_2, ...,s_t$. So the equation will have the form:

$$L= v_1s_1+ v_2s_2 +...,+ v_ts_t.$$

Once this equation is obtained, the success or failure of a session will be established on the basis of the value of L obtained for that session.

$$If\begin{cases} L \geq 0 \text{ session success is predicted} \\ L < 0 \text{ se session failure is predicted} \end{cases}$$

Discriminant analysis is useful in finding the most relevant semantic concepts. In each step of the stepwise discriminant technique the importance of each attribute included can be studied. This is important not only because we will have the equation to estimate the value of a session in the future but also because the method provides the analyst with a criteria to understand which actions are most relevant for the success of a session from each viewpoint. The computation of the discriminant function has been done according to [18, 2, 1, 24]
The model estimates the coefficients (values) of each attribute considered (semantic) for each pair of goals and viewpoint considered.

From this result it is possible to establish the relationship between semantics and the result of the session. Those coefficients with higher positive value are associated with sessions ending successfully and those taking negative values are associated with failure sessions. So it would be desirable that pages visited by users in their sessions have semantics related to these coefficients associated with them.
So the innovative aspect of the proposed approach has to do with both the predictive model and the measure used to establish concepts or actions that happen during a session to make it successful from a certain site viewpoint. Notice that the number of relevant concepts (*N1)* will be much less than the number of pages *N (N1<<N)* in a web site, so that the problem of analyzing user session decreases in complexity, improving the performance of the methods used to analyze them.

4. Architecture Overview

In order to deploy the methodology described above, it is necessary to define a global architecture, including the modules that deal with the method tasks. The main tasks are:

1. Preprocessing: The result of this task is the Enriched Sessions table.
2. Classification: The result of this task is the Results table.
3. Usage of the predictive model.
4. Refining stage: This task adjusts the model to new results.

Web Mining tasks have been often implemented by agents. The agent paradigm offers desirable features such as autonomy, that is, the ability of acting itself and on behalf of others and proactivity, that is, the ability of acting in anticipation of future problems, needs or changes. These characteristics are very suitable in the scenario described in previous sections because of its dynamic idiosynchrasy.

Thus, a multiagent architecture is proposed, which is composed of three different layers:

1. Semantic Layer: This layer contains agents related to the logic of the algorithm or method used in the ongoing session value estimation.
2. Optimization/Decision Layer: This layer contains agents responsible for optimizing or taking decisions depending on the estimated value.
3. Services Provider Layer: This layer contains agents that provide several services to the rest of the agents. These services are generic and useful tasks, which are independent of the other layers and offer an interface, which may be used by any agent who asks for a service.

Semantic Layer

This level is fed directly by the session value estimation implementation. The agents of this layer deal with the concepts value estimation and its usage in the following sessions. This layer is a multiagent subsystem composed of different specialized agents. They are:

- Preprocessing agents: These agents are responsible for enriching the sessions.
- Classification agents: Although the classification of a session as success or failure is made according to an expert, it is possible to automate this task, through the usage of previous classifications and expert knowledge.
- Estimation agents: These agents apply the stepwise multivariate predictive model. In a first phase, this operation is made offline. Nevertheless, the algorithm must be applied online in current web usage data to estimate the result of a session.
- Refining agents: If the prediction of the used model is not successful, it is necessary to refine the algorithm with new information. This kind of agents must communicate with the estimation agents in order to inform them about the changes.

As we can see, using agents in the semantic layer provides dynamism to the algorithm deployment.

Optimization/Decision-making Layer

This layer includes agents that make decisions or optimize depending on the information supplied by the semantic layer. Although it is possible to build other kind of agents, we have defined the following agents with the aim of optimizing the accesses and personalizing usage of the web:

- Prefetching agents: These agents prefetch most probably next visited web pages, depending on the session value, that is, giving priority to the sessions with a higher value. In this way, the web session load is more efficient and the user feels more comfortable with the web site.
- Adaptive agent: These agents are responsible for building offers adapted to the preferences of the users. These offers may show as popups or web pages. Any other kind of personalization may be added to the logic of these agents.

Services Provider Layer

This level includes generic services used for assisting other layers agents. These agents are:

- Data retrieval agents: These agents goal is to retrieve data from different information sources. These sources are heterogeneous (i.e., databases, files) and they have different types of information and access requirements. Therefore, these agents may delegate on specialized agents for different sources.
- Locator agents: These agents aim to connect and communicate an agent with another one. For finding a given agent, the locator agents use its category, that is, the type of agent. If there are no available agents of this type, the system launches a new agent for serving this request.

It is possible to add more services, implementing the corresponding agent and defining an interface with the rest of the architecture.

Figure 1 represents the three layers of the proposed architecture and their relationships. Notice that there are both internal and external relationships among different kinds of agents. The different information sources are also shown in the graphic.

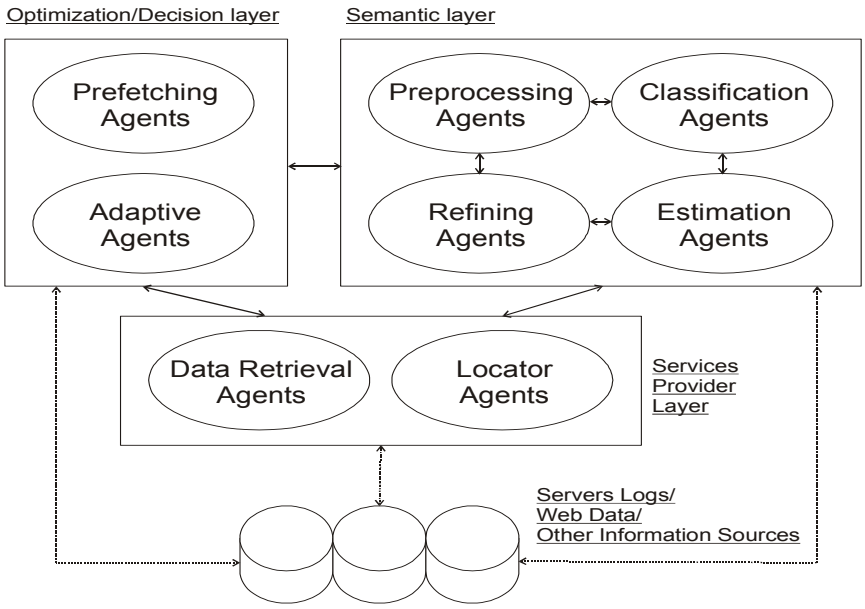


Fig. 1: Web-behavior agent-based architecture layers

5. Case Study - Experimental Results

In order to validate the proposed approach we have applied it to an e-commerce site with 2500 pages. We have preprocessed the log data from accesses to the server during a period of one month. After filtering out irrelevant entries, the data were segmented into 38058 sessions. We considered 20 goals, five points of view and 94 semantic actions. We are dealing with an online shop in which a set of historical sessions has been given that have been classified according to different viewpoints. As an example of these different viewpoints, the success criteria for one department (marketing) is that the user acquires a product while for another department (sales) the success criteria is given by the amount of benefit of the visit (amount sold).

The main goal of the analysis is to estimate the result of a session. Additional benefits of the proposed approach are:

- Establishment of the concepts or actions behind the visited pages (semantics) that are more relevant for the success of a session (i.e., those that contribute most to the site achievement of goals).
- To provide on-line information to automatically make decisions on what motivates the user (e.g., automatic precatching, online offers and discounts).

For each goal and viewpoint of the site, the value of each concept has been established and the results shown in table 2 have been obtained. Notice that it is a good model as it predicts the 93,3% of the successful sessions and the 82,2% of the failure sessions.

	Predicted	
Observed	Success	failure
Success	93,3	6,7
failure	17,8	82,2

Table 2: Summary discriminant table

The results for different viewpoints and goals considered in the experimental results are shown in table 3.

		Viewpoint 1				2			
GOAL		Predicted				Predicted			
1	Observed	Success	Failure	Total	Observed	Success	Failure	Total	Observed
	success	93,3	6,7	100	Success	94,5	5,5	100	success
	Failure	17,8	82,2	100	Failure	14,3	85,7	100	Failure
		Predicted				Predicted			
2	Observed	Success	Failure	Total	Observed	Success	Failure	Total	Observed
	success	85,4	14,6	100	success	87,7	12,3	100	success
	Failure	12,5	87,5	100	Failure	14,6	85,4	100	Failure
		Predicted				Predicted			

Table 3 : Sample of Summary tables

The discriminant function obtained was:

$$L=0,431\ s_1+,0,99\ s_1+\ 0\ ,126\ s_2+0,751s_3-0,444\ s_4-0,107\ s_5+0,119\ s_6+0,742\ s_7+0,306s_8$$

The proposed analysis made it possible to establish the relevant factors to take into account when analyzing sessions. Eight relevant concepts to estimate session results were obtained from the 94 initial number of concepts.
In Figure 2, it is shown (Table 2) that most of the sessions were successfully classified. The in-depth analysis of the wrong classified sessions (Figure 3) helped the site sponsor recognize market niches. Note that it is more difficult to describe the wrongly classified examples as they are found in the edges.

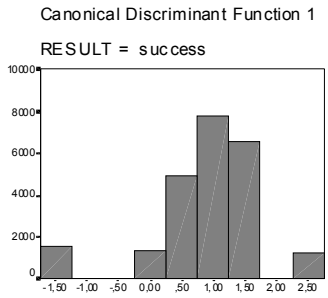


Fig. 2: Canonical Discriminant Function 1 - RESULT = success

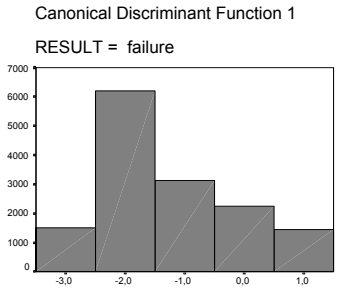


Fig. 3:Canonical Discriminant Function 1 - RESULT = failure

In 83% of the combinations of goals and viewpoints the number of relevant concepts able to classify a session was 20 % or less of the 94 initial concepts.
Finally, when goals were weighted according to each point of view taken into account, an aggregate perception of all goals was obtained for each point of view (i.e., the same session was successful for the Department of Marketing but not for the Sales Department).

6. Conclusions

In this paper we have presented an approach based on discriminant analysis that helps establish semantic factors (e.g., actions, elements) that are relevant for the success of a session. The methodology is also useful for classifying on-going sessions so that actions can be taken to motivate the user to stay longer in the site or to leave it. An innovative aspect of the approach is that the semantics of the pages is established according to different goals and viewpoints so that the analysis can be done according to different criteria. Besides, an agent-based architecture has been defined with the aim of providing dynamism to method deployment. The proposed approach has been applied to an e-commerce site and the results not only helped establish on-line discounts and offers to visitors but also helped the site sponsors discover some market-niches.
New challenges and future research directions address the problem of classifying sessions as success or failure and of having a measure of such success or failure. We are also working on a mechanism to automate this task. On the other hand, we are also analyzing how the sequence of events can be determinant in the success of the session. These open issues could be developed and addressed by multiple alternatives and the forecoming work on this approach will present them.

Acknowledgments

The research has been partially supported by Universidad Politécnica de Madrid under Project WEB-RT and Programa de Desarrollo Tecnológico (Uruguay).

References

1. R. E. Anderson, R. L. Tatham, W. C. Black. Multivariate Data Analysis (5th Edition) J. F. Hair (Editor), Prentice Hall College Div; 5th edition, March 1998
2. T. W. Anderson. An Introduction to Multivariate Statistical Analysis, 2nd Edition Wiley-Interscience; 2 edition, September 1984
3. S. Ansari, R. Kohavi, L. Mason, Z. Zheng. Integrating E-commerce and Data Mining: Architecture and Challenges. In Workshop on Web Mining for E-Commerce - Challenges and Opportunities Working Notes (KDD2000), pages 27-34, Boston, MA, August 2000
4. B. Berent. Detail and Context in Web Usage Mining: Coarsening and Visualizing Sequences, In R. Kohavi, B.M. Masand, M. Spiliopoulou, & J. Srivastava (Eds.), *WEBKDD 2001 - Mining Web Log Data Across All Customer Touch Points* (pp. 1-24). Berlin. Springer, LNAI 2356.
5. B. Berendt. Using Site Semantic to Analyze, Visualize, and Support Navigation. Data Mining and Knowledge Discovery 6, pp.37-59, 2002.
6. B. Berendt, A. Hotho, G. Stumme. Towards Semantic Web Mining. ISWC 2002, LNCS 2342, pp. 264–278, 2002.
7. B. Berendt, M. Spiliopoulou. Analysis of navigation behavior in web sites integrating multiple information systems. The VLDB Journal, 9. pp.56-75, 2000.
8. P. Berthon, L. Pitt, R. Watson. The World Wide Web as an advertising medium. Journal of Advertising Research 36(1) pp.43-54,1996
9. J. Borges, M. Levene. A heuristic to capture longer user web navigation patterns. In Proc. of the First International Conference on Electronic Commerce and Web Technologies, Greenwich, U.K., September 2000.
10. J. Broekstra, A. Kampman, F. Harmelen. Sesame: An Architecture for Storing and Querying RDF Data and Schema Information , Semantics for the WWW, MIT Press, 2001, D. Fensel, J. Hendler, H. Lieberman and W. Wahlster.
11. E. H. Chi, P. Pirolli, J. Pitkow. The Scent of a Site: A System for Analyzing and Predicting Information Scent, Usage, and Usability of a Web Site. In Proc. of CHI '2000 The Hague, Amsterdam CHI Letters volume 2, issue 1.
12. J. Collins, W. Ketter, M. Gini, B. Mobasher. A Multi-Agent Negotiation Testbed for Contracting Tasks with Temporal and Precedence Constraints" Int. Journal of Electronic Commerce, 7(1):35--57, 2002.
13. H. Dai, B. Mobasher. A Road map to More Effective Web Personalization: Integrating Domain Knowledge with Web Usage Mining. Proc.of the International Conference on Internet Computing 2003 (IC'03), Las Vegas, Nevada, June 2003.
14. R. Doorenbos, O. Etzioni, D. Weld. *A Scalable Comparison-Shopping Agent for the World-Wide Web*. In Proceedings of Autonomous Agents (Agents'97), 1997.
15. J. Eighmey. On the Web: It's What You Say and How You Say It. Greenlee School of Journalism and Communication, Iowa State University. [Eighmey Website Response Profile. http://eighmey.jlmc.iastate.edu/DE.html].
16. D. Florescu, A. Levy, A. Mendelzon. Database techniques for the World-Wide Web: A survey. SIGMOD Record (ACM Special Interest Group on Management of Data), 27(3):59.

17. I. Horrocks, S. Tessaris. Querying the Semantic Web: A Formal Approach. ISWC 2002, LNCS 2342, pp. 177–191, 2002.
18. D. G. Keibaun, L. L. Kupper, K. E. Muller Applied Regression Analysis and Other Multivariable Methods. PWS-KENT Publishing Company, USA 1988.
19. A. Magkanaraki, S. Alexaki, V. Christophides, D. Plexousakis. Benchmarking RDF Schemas for the Semantic Web. ISWC 2002, LNCS 2342, pp. 132–146, 2002.
20. E. Menasalvas, S. Millán, E. Hochsztain. A Granular Approach for Analyzing the Degree of Afability of a Web Site. In International Conference on Rough Sets and Current Trends in Computing RSCTC2002. October 2002.
21. B. Mobasher, H. Dai, T. Luo, M. Nakagawa, J. Witshire. Discovery of aggregate usage profiles for web personalization. In Proceedings of the WebKDD Workshop. 2000.
22. Mobasher B., Dai H., Luo T., Sun Y., Zhu J. Integration Web Usage and Content Mining for More Effective Personalization. In Proceedings of the International Conference on E-Commerce and Web Technologies (ECWeb2000), September 2000, Greenwich, UK.
23. B. Mobasher, N. Jain, E. Han, J. Srivastava. Web mining: Pattern discovery from WWW transaction. In Int Conf. on Tools with Artificial Intelligence, pp 558-567, New Port, 1997.
24. D. Morrison. Multivariate Statistical Methods, Duxbury Press; 4th edition. 2002.
25. O. Nasraoiu, R. Krisnapuram, A. Joshi. Mining web access logs using a fuzzy relational clustering algorithm based on a robust estimator, 1998.
26. D. Oberle, B. Berendt, A. Hotho, J. Gonzalez. Conceptual User Tracking. In Advance in Web Intelligence. LNAI 2663. AWIC 2003, Madrid, pp. 155-164, May 2003.
27. M. Papazoglou. Agent-Oriented Technology in Support of e-business. Communications of the ACM, Vol 44, N. 4. 2001.
28. T. R. Payne, R. Singh and K. Sycara. Browsing Schedules - An Agent-Based Approach to Navigating the Semantic Web. I. Horrocks and J. Hendler (Eds.): ISWC 2002, LNCS 2342, pp. 469–473, 2002. Springer-Verlag Berlin Heidelberg 2002.
29. J. Pei, J. Han, B. Mortazavi-Asl, H. Zhu. Mining access patterns efficiently from web logs. In Proceedings Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'00), 2000.
30. B. Sarwar, J. A. Konstan, A. Borchers, J. Herlocker, B. Miller, J. Riedl. *Using Filtering Agents to Improve Prediction Quality in the GroupLens Research Collaborative Filtering System*. CSCII'98 , Seattle WA, 1998.
31. C. Shahabi, A. M. Zarkesh, J. Adibi, V. Shah. Knowledge discovery from user's web-page navigation. In Proceedings of the Seventh International Workshop on Research Issues in Data Engineering, High Performance Database Management for Large-Scale Applications (RIDE'97), pages 20-31, Washington- Brussels - Tokyo, IEEE. 1997.
32. A. Sieg, B. Mobasher, S. Lytinen, R. Burke. Concept Based Query Enhancement in the ARCH Search Agent. Proceedings of the 4th International Conference on Internet Computing (IC'03), Las Vegas, June 2003.
33. M. Spiliopoulou, C. Pohle. Data Mining for Measuring and Improving the Success of Web Sites. Data mining and Knowledge Discovery, (5)1-2, 2001.
34. M. Spiliopoulou, C. Pohle, L. Faulstich. Improving the effectiveness of a web site with web usage mining. In Proceedings WEBKDD99, 1999.
35. J. Srivastava, R. Cooley, M. Deshpande, P. Tan. Web usage mining: Discovery and applications of usage patterns from web data. SIGKDD Explorations, 1:12-23, 2000.
36. Y. Sure, M. Erdmann, J. Angele, S. Staab, R. Studer, D. Wenke. OntoEdit: Collaborative Ontology Development for the Semantic Web. ISWC 2002, LNCS 2342, pp. 221–235.
37. G. Tewari, P. Maes. Design and Implementation of an Agent-Based Intermediary Infrastructure for Electronic Markets. *EC'00*, October 17-20, 2000, Minneapolis.
38. G. Wolfang, S. Lars. Mining web navigation path fragments. In Workshop on Web Mining for E-Commerce - Challenges and Opportunities Working Notes (KDD2000), pages 105–110, Boston, MA, August 2000.

Greedy recommending is not always optimal

Maarten van Someren[1], Vera Hollink[1] and Stephan ten Hagen[2]

[1] Dept. of Social Science Informatics, University of Amsterdam,
Roetersstraat 15, 1018 WB Amsterdam, The Netherlands,
{maarten,vhollink}@swi.psy.uva.nl
[2] Faculty of Science, University of Amsterdam,
Kruislaan 403, 1098 SJ Amsterdam, The Netherlands
stephanh@science.uva,nl

Abstract. Recommender systems help users to find objects or documents on web sites. In many cases it is not easy to know in advance by whom and for what purpose a web site will be used. This makes it difficult for many applications to define adequate recommendations in advance. Therefore recommendations are typically generated dynamically. Recommendations are based on analysis of user data (social filtering) or content (content-based) or a mixture of these. Current methods optimise the quality of recommended objects (e.g. the probability that it is the target of the user, or the estimated interestingness). Even with recommendations, many steps are often needed to locate the desired information. This changes the task to what we call "sequential recommending": a series of recommendations in which the user indicates his preference, leading to a target object. Here we argue that in sequential recommending a series of normal, "greedy", recommendings is not always the strategy that minimises the number of steps in the search. Greedy sequential recommending conflicts with the need to explore the entire space and may lead to recommending series that require more steps (mouse clicks) from the user than necessary. We illustrate this with an example, analyse when this is so and outline a more efficient recommendation method.

1 Introduction

Recommender systems typically recommend one or more objects that appear to be the most interesting for the current user (e.g. [2–6, 12]). These systems take information about the current context (documents, screens, actions) that were selected by the user and use this to recommend further objects. In practice, many recommending tasks need more than one recommendation step. For example searching for information in an interactive information system typically takes a number of steps. Even if the information system provides recommendations at each step, a number of steps will be needed. This has consequences for the recommending task and the applicable methods. During the interaction, the recommender acquires information about the users goal. This information can be used to generate better recommendings than can be found from the direct context alone. We shall call recommending in which the recommender presents objects it predicts to be the target or closest to the target *greedy* recommending. If it takes information about the user (like the interaction history) into account we call it *user-adaptive greedy recommending*. If recommending takes place over a series of interaction steps

that ends with finding a target object, we call it *sequential recommending* in contrast to *one-step recommending*. Most recommendation methods use a form of collaborative filtering (recommending objects which were targets of similar users) or content based filtering (recommending objects which are similar to objects which were positively evaluated before) or a combination of these techniques. In this paper we mainly consider recommender systems which only use collaborative filtering.

In this paper we note two problems of greedy recommending. The first problem is the *inadequate exploration problem*. The recommender needs data about users preferences. A recommender system generates recommendations but at the same time it has to collect data about user preferences. Greedy recommending may have the effect that some objects are not seen by users and therefore are not evaluated adequately. This may prevent popular objects from being recommended. In section 3 we address this problem and show how exploration can be integrated in a recommender system.

The second problem is that in sequential recommending settings, greedy recommending may not be the most efficient method for reaching the target. In a setting in which the user is looking for a single target object, a criterion for the quality of recommending is the number of links that needs to be traversed to reach the target. We can view recommending as a classification task where the goal is to 'assign' the user to one of the available objects in the minimal number of steps. It is intuitively clear that "greedy sequential recommending", presenting the most likely target objects, may not be the optimal method. For this setting, a more efficient method is binary search: using information about the user that makes it possible to eliminate half of the candidate target objects.

This means that to minimise the length of the path to the most interesting object it is better to start recommending objects with a lower probability of being the user's target, but with a higher informational value. For example, if two operas are the most popular objects on a site that recommends music, it may still be better to recommend one prototypical opera and one prototypical rock song.

In section 4 we address the second problem. We propose a new strategy based on binary search and show in an example that this strategy can *under certain circumstances* improve the results of greedy recommending. The last section contains conclusions and suggestions for further research.

2 The inadequate exploration problem

Recommenders based on "social filtering" need data about users. Unfortunately, acquiring the data about user preferences interferes with the actual recommending. There is a conflict between "exploitation" and "exploration" (e.g. [9]). In [11] we showed that greedy recommenders might get stuck in a local optimum and never acquire the optimal recommendation strategy. The possible paths through the site which the user can take are determined by the provided recommendations. The user is forced to click on one of the recommended objects even when his target object c_t is not among the recommendations. The system observes which object is chosen and infers that this object was indeed a good recommendation. It increases the chance that the same object is recommended again in the next session and the system never discovers that c_t would have been an

even better recommendation. Objects that are recommended in the beginning will become popular because they are recommended, but highly appreciated objects with a low initial estimated appreciation might never be recommended and the system sticks with a suboptimal recommendation strategy.

To make sure the estimation of all content objects becomes accurate it is necessary to explore the entire preference space. The system always has to keep trying all objects even if the user population is homogeneous. One way to perform this kind of exploration, is to use a ϵ-greedy method [8]. Here the assumed optimal objects are recommended with probability 1-ϵ and a random other object with probability ϵ. By taking ϵ small, the system can make good recommendations (exploitation), while assuring all objects will eventually be explored. Note that this agrees with the empirical observations in [10], where it is suggested that recommendations should be reliable in the sense that they guide the users to popular objects. But also new and unexpected objects should be recommended to make sure that all objects are exposed to the user population.

3 Suboptimality of greedy sequential recommending

Greedy recommending may not be optimal for sequential settings. In this section we analyse the performance of three recommending methods. To enable this analysis we define a specific recommending setting.

3.1 Setting

Although recommending is a single term for the task of supporting users by recommending objects from a large repository, there is actually a wide range of recommendation tasks. We focus on one particular recommendation setting to compare the effects of three recommendation strategies. We keep the setting as simple as possible to show the differences between different recommendation strategies but the arguments still hold for more realistic situations. The setting we use has the following properties:

- The recommender recommends elements from a fixed set of content objects: $\{c_1, \cdots, c_n\}$.
- Every user is looking for one particular *target* content object, but this is not necessarily the same object for each user.
- In each cycle, the system recommends exactly two content objects to the user.
- After receiving a recommendation the user indicates for one of the objects *"My target is X."* or *"X and Y are both not my target, but object X is closer to what I am looking for than Y."*.
- If the user has found his target then the interaction stops else he receives two new recommendations.

In this setting the main task of the recommender is to find at each step two content objects to recommend.

3.2 Three recommenders

We distinguish three approaches to sequential recommending:

- Non-user-adaptive recommending
- Greedy user-adaptive recommending
- Exploratory user-adaptive recommending

Since these methods rely on data about the preferences of users, an important aspect of the recommendations task is to acquire these data. Non-user-adaptive recommender systems only need statistical data about the user population as a whole. Methods which adapt to individual users also need to keep track of the preferences of the current user of the site. In the following sections we will discuss the advantages and disadvantages of each of these strategies. Specifically, we specify when *greedy user-adaptive recommending* is better than *non-user-adaptive recommending* and when *exploratory user-adaptive recommending* is better than *non-user-adaptive recommending*.

The analysis uses the following scheme. At each step, the recommender presents two objects. The user may accept one of these or may prefer one over the other, after which the recommender presents two new objects. At each presentation there is a probability that the target was presented, resulting in 0 additional presentations. If the user does not accept the presented objects then these are excluded and recommending is applied to the remaining objects. In case of binary choice, we can estimate the number of steps to the target as the amount of information:

$$I = \sum_i P_i \log_2 P_i \tag{1}$$

Here i ranges over all objects. This can be interpreted as the minimal number of bits that is needed to identify the target. It approximates (nearly) optimal strategies and gives an optimistic estimate for poorer strategies. The effect of presenting an object to the user on the expected path length can be expressed as:

$$(P_{\mathrm{object1}} + P_{\mathrm{object2}}) \cdot 0 + (1 - P_{\mathrm{object1}} - P_{\mathrm{object2}}) \sum_i P_i \log_2 P_i \tag{2}$$

Here the first term denotes the probability that the user accepts object 1 or 2. If the user accepts one of the two presented objects then no more steps are needed. The second term consists of the remaining probability and its estimated path length.

Our main point is to demonstrate that greedy recommending is not always the optimal method for this problem. We base the discussion on the simple case of a domain in which objects can be scaled on a single dimension (see [1] for an overview of scaling methods). One-dimensional scaling methods take a (large) number of ratings or comparisons of objects by different persons as input and define an ordering such that each person can be assigned a position on the scale that is consistent with his ratings or comparisons. In our case we are dealing with comparisons acquired during earlier recommending sessions that need to be acquired during a learning phase.

It is often not possible to construct a perfect scale. Many domains have an underlying multidimensional structure. In this case a multi-dimensional scaling method can be

applied. Here we restrict the discussion to the one-dimensional case because our goal is just to demonstrate the principle. Once items are scaled, persons can be given a position on the scale that corresponds to their favourite object. This means that we also obtain a probability distribution over the scaled items.

3.3 Non-user-adaptive sequential recommending

The first recommendation strategy that we will discuss is non-user-adaptive recommending. This is a greedy method that does not use information about individual users. It recommends objects ordered by the marginal probability that the object is the target. This strategy is often implicit in manually constructed web sites. The designer estimates which objects are most popular and uses this estimate to order the presentation of objects to the user [7]. A recommender that uses this strategy needs data to estimate for each object the probability that it is the target.

We can estimate the number of steps that the *non-user-adaptive sequential recommender* needs with equation (2). For this recommender method, P_{object1} and P_{object2} are at each step the probabilities of the objects with highest marginal probability that were not presented before.

3.4 Greedy user-adaptive sequential recommending

User-adaptive recommenders collect information about the current user during a session and use this to generate *personalised* recommendations. In our setting the only available data about the preferences of a user is a series of rejected objects and preferences that come out of interaction logs. A *greedy* user-adaptive recommender system always recommends the objects with the highest probability of being the target object *given the observed preferences*. If the user has indicated a preference of c_1 over c_2 and c_3 over c_4 et cetera, a greedy user-adaptive recommender recommends c_i with maximal $P(c_i = \mathrm{target} | c_1 > c_2 \,\&\, c_3 > c_4 \,\&\, \ldots)$. If the preference of one object changes the probability that the other object is the target, this strategy can reduce the expected path length compared to non-user-adaptive recommending as illustrated by the following example. Suppose a recommender system recommends pieces from a music database consisting of operas and pop songs and in the first cycle the system has recommended the opera 'La Traviata' and the song 'Yellow Submarine'. If the user indicates that he prefers 'La Traviata' over 'Yellow Submarine', it becomes more probable that the user is looking for an opera than a pop song, even if more people ask the database for pop songs. In the next step a user-adaptive recommender would use this information and recommend two operas. A non-user-adaptive recommender system on the other hand would recommend the same two objects as when the user had chosen the pop song.

This recommender method also needs data to estimate the conditional probabilities of all objects or to predict the best object. Obviously, estimating the conditional probabilities needs far more data than the marginal probabilities. Like the previous method, this second method is greedy, because it always recommends the object with the highest estimated conditional probability of being the target. In the example above, this means a greedy user-adaptive recommender would recommend the two most popular operas.

For this method we get the following expression for the expected path length:

$$(1 - P_{\text{best}}) \cdot \sum_i P_i \log_2 P_i - P_{\text{next best}} \cdot (\sum_{i<mid} P_i \log_2 P_i + \sum_{i>mid} P_i \log_2 P_i) \quad (3)$$

Here mid is the point between *best* and *next best*. The difference with the previous method is that the *greedy user-adaptive sequential recommending* selects *best* and *next best* on the basis of the users preferences indicated in earlier presentations instead of (only) on the basis of marginal probabilities. This means that P_{best} and $P_{\text{next best}}$ will be higher at each step and the third term will be lower because the probabilities P_i are likely to vary more from 0.5.

4 Exploratory sequential recommending

The third recommending method is based on the idea that sequential recommending can be viewed as a kind of classifi cation, the task is to assign the user to the object that matches his interest, and is based on binary search. Exploratory sequential recommending consists of repeating the following steps until the target has been found:

1. Find a point such that the sum of probabilities objects on both sides is equal: the "center of probability mass".
2. Within each subset, fi nd the object with the maximum probability of being the target.
3. Present these to the user as recommendations.
4. The user now indicates which of the presented objects he prefers and whether he wants to continue (if neither of the presented objects is the target).
5. If neither of the presented objects is accepted then eliminate *all* objects on the side of the midpoint where the least-preferred object is located.

This is a form of standard binary search. It needs a way to eliminate the objects that are closer to one presented object than to the other. If this is available then we get for the expected path length the following:

$$(1 - P_{\text{best}}) \cdot \sum_i P_i \log_2 P_i - P_{\text{symmetrical}} \cdot \sum_{i<mid} P_i \log_2 P_i \quad (4)$$

or

$$(1 - P_{\text{best}}) \cdot \sum_i P_i \log_2 P_i - P_{\text{symmetrical}} \cdot \sum_{i>mid} P_i \log_2 P_i \quad (5)$$

where the choice depends on the users preference. Here *symmetrical* refers to the object that is at equal distance from the center of probability mass but here the sum ranges only over the objects that carry half the probability mass. In the last term i ranges over *either* the part above *or* under the "center of probability mass". This expression clearly shows the difference with the greedy method: the fi rst term $1 - P_{\text{best}}$ is equal for both methods because they both present the object with the highest probability of being the target. The second term is on average smaller for the exploratory method because it is the next most likely object *from a subset* and not from the whole set, as in the greedy method, where P_i effectively ranges over all i.

5 Comparison of the methods

In terms of equation 1, the methods differ in the probability that the target is presented and in the expected length of the remaining steps. Here our goal is to demonstrate that "greedy" methods are not optimal in the sense that they do not always lead to recommendations that minimise the number of steps until the target is found. Intuitively, greedy methods maximise the probability of an immediate hit and neglect the expected length of the remaining path where non-greedy methods minimise the total expected path length!

We illustrate this with an example. Suppose in a domain objects can be scaled perfectly on a single dimension. The probability distribution of objects being a target of the user is skewed. For argument sake we take a linear function, starting with the most popular object (highest probability) and ending with the object with the lowest probability. In this case, both the *non-user-adaptive sequential recommender* and the *greedy user-adaptive sequential recommender* will recommend the objects starting at the highest marginal probability and following the scale down until the object with the smallest marginal probability.

The *exploratory sequential recommender* will recommend the objects with the highest marginal probability but the second object will not be the second best but the one with the lowest marginal probability! In terms of equation (1) we get the following estimates for the number of items that need to be presented: *Non-user-adaptive sequential recommender*:

$$(P_{\text{best}} + P_{\text{next best}}) \cdot 0 + (1 - P_{\text{best}} - P_{\text{next best}}) \cdot \sum P_i \log_2 P_i \tag{6}$$

If we disregard the first term and the common part of the second term, we get:
Greedy user-adaptive sequential recommender:

$$(1 - P_{\text{best}}) \cdot \sum P_i \log_2 P_i - P_{\text{next best}} \cdot \sum P_i \log_2 P_i \tag{7}$$

Exploratory sequential recommender:

$$(P_{\text{best}} + P_{\text{symmetrical}}) \cdot 0 + (1 - P_{\text{best}} - P_{\text{symmetrical}}) \cdot \sum P_i \log_2 P_i \tag{8}$$

Consider what happens in the first step. The difference in expected path length after one presentation is the difference between:

$$\log_2 P_i - P_{\text{next best}} \cdot \sum P_i \log_2 P_i \tag{9}$$

and

$$\log_2 P_j - P_{\text{symmetrical}} \cdot \sum P_j \log_2 P_j \tag{10}$$

This shows that in qualitative terms, *exploratory sequential recommender* is going to be better if $P_{\text{symmetrical}}$ is larger than $P_{\text{next best}}$ and the average remaining path length is smaller. This happens when differences in marginal probability are smaller and the number of objects is larger. In the extreme case of a uniform distribution the *non-user-adaptive sequential recommender* cannot choose between objects. The behaviour

of the *greedy user-adaptive sequential recommender* and the *exploratory sequential recommender* is as in the skewed-distribution example, but the probability of being the target will be small for (*best* and) *next best* and the remaining path length term large. The path length of the *exploratory sequential recommender* is likely to be smaller than of the *greedy user-adaptive sequential recommender*.

When data or prior knowledge are available on the (conditional) probabilities that an object is the target then the formulae can be used to predict the effect of different methods on the expected path length.

This illustration uses a set of objects that can be perfectly scaled on a single dimension. This is of course an exception. Scales will not perfect and many problems involve more dimensions. In this case a more general approach is needed. This is provided by multi-dimensional scaling, which amounts to a form of clustering. Instead of using feedback on recommended objects to eliminate part of a one-dimensional scale, we use feedback to eliminate half of the clusters.

6 Discussion

Our analysis of sequential recommending shows the following:

- For any recommender system it is necessary to not immediately start recommending but to make sure that the entire space is presented to users and evaluated. In practice this can be achieved in different ways, for example by including a random component in the recommender or by systematically exploring the space in the initial stage of the application. In this case the user should understand what the "recommender" is doing (and that its goal is ultimately to help the user community).
- Greedy recommending is not always the method that finds the target in the minimal number of interactions (or mouse clicks). Exploratory recommending can be shown to be better under certain conditions that can be specified. It seems that when recommending sequences are short, the difference between greedy and exploratory recommending is small. Large sites can benefit most from exploratory recommendations because in each step the set of potential target objects is reduced more than for greedy recommending. This prevented that unpopular objects become unreachable by requiring too many mouse clicks.

There are a number of issues that need further work. One is the combination with content-based methods. These can be used to model the space in which sequential recommending works and it can be used alone or together with the scaling approach above by the exploratory recommender. Other issues are different forms of recommending. Here we restricted the discussion to a specific recommender setting. We believe that that the principle used above is also relevant for other recommending settings, although details may be different. For example, if more than two objects are presented application of the binary search principle is more complicated and if ratings are used, a different method is needed but the approach remains the same.

References

1. C. Coombs. *A Theory of Data*. John Wiley, New York, 1964.
2. A. Kiss and J. Quinqueton. Multiagent coöperative learning of user preferences. In *Proceedings of the ECML/PKDD Workshop on Semantic Web Mining 2001*, pages 45–56, 2001.
3. P. Melville, R.J. Mooney, and R. Nagarajan. Content boosted collaborative filtering. In *Proceedings of the SIGIR-2001 Workshop on Recommender Systems*, 2001.
4. A. Moukas. User modeling in a multiagent evolving system. In *Proceedings of the ACAI'99 Workshop on Machine learning in user modeling*, pages 37–45, 1999.
5. M. Perkowitz and O. Etzioni. Adaptive web sites: Automatically syntasizing web pages. In *Proceedings of AAAI98*, pages 727–732, 1998.
6. I. Schwab and W. Pohl. Learning user profiles from positive examples. In *Proceedings of the ACAI'99 Workshop on Machine learning in user modeling*, pages 21–29, 1999.
7. T. Sullivan. Reading reader reaction: A proposal for inferential analysis of web server log files. In *Proceedings of the Human Factors and the Web 3 Conference, Practices & Reflections*, 1997.
8. R.S. Sutton. Generalizing in reinforcement learning: Succesful examples using sparse coarse coding. In *Advances in Neural Information Processing Systems (8)*, 1996.
9. R.S. Sutton and A.G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.
10. K. Swearingen and R. Sinha. Beyond algorithms: An hci perspective on recommender systems. In *ACM SIGIR 2001 Workshop on Recommender Systems*, New Orleans, Lousiana, 2001.
11. S. ten Hagen, M. van Someren, and V. Hollink. Exploration/exploitation in adaptive recommender systems. In *To appear in proceedings of Eunite 2003*, 2003.
12. T. Zhang and V.S. Iyengar. Recommender systems using lineair classifiers. *Joural of Machine Learning Research*, 2:313–334, 2002.