ECML/PKDD 2012

EUROPEAN CONFERENCE ON MACHINE LEARNING
AND
PRINCIPLES AND PRACTICE OF KNOWLEDGE DISCOVERY IN
DATABASES

The Third International Workshop on Mining Ubiquitous and Social Environments

MUSE'12

September 24, 2012

Bristol, UK

Editors:
Martin Atzmueller
Knowledge and Data Engineering Group, University of Kassel
Andreas Hotho
Data Mining and Information Retrieval Group, University of Wuerzburg

Table of Contents

Preface

The 3^{rd} International Workshop on Mining Ubiquitous and Social Environments (MUSE 2012) was held in Bristol, UK, on September 24^{th} 2012 in conjunction with the The European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD 2012).

The emergence of ubiquitous computing has started to create new environments consisting of small, heterogeneous, and distributed devices. These foster the social interaction of users in several dimensions. Mining in ubiquitous and social environments is an established area of research focusing on advanced systems for data mining in such distributed and network-organized systems. However, the characteristics of ubiquitous and social mining are in general rather different from common mainstream machine learning and data mining approaches: The data emerges from potentially hundreds to millions of different sources, in a collective way. Since these sources are usually not coordinated, they can overlap or diverge in any possible way.

In typical ubiquitous settings, the mining system can be implemented inside the small devices and sometimes on central servers, for real-time applications, similar to common mining approaches. Therefore, the analysis of the collected data, the adaptation of well-known data mining and machine learning approaches, and finally the engineering and development of new algorithms are challenging and exciting steps.

The goal of this workshop is to promote an interdisciplinary forum for researchers working in the fields of ubiquitous computing, social web, Web 2.0, and social networks which are interested in utilizing data mining in a ubiquitous setting. The workshop presents contributions adopting state-of-the-art mining algorithms on ubiquitous social data, ranging from the analysis of social networks and ubiquitous environments to applications combining both, for example, supporting people in their daily lives using smart city services. This is implemented by combining the data from both worlds - human activies measured by sensors and manifested in social networks - in the mining process. With this workshop, we want to accelerate the process of identifying the power of advanced data mining operating on data collected in such ubiquitous and social environments.

We would like to thank the invited speaker, all the authors who submitted papers and all the workshop participants. We are also grateful to members of the program committee members and external referees for their thorough work in reviewing submitted contributions with expertise and patience. A special thank is due to both the ECML PKDD Workshop Chairs and the members of ECML PKDD Organizing Committee who made this event possible.

Bristol, September 2012

Martin Atzmueller
Andreas Hotho

Workshop Organization

Workshop Chairs

Martin Atzmueller	University of Kassel - Germany
Andreas Hotho	University of Würzburg - Germany

Program Committee

Ulf Brefeld	Yahoo Research - Spain
Ricardo Cachucho	Leiden University - The Netherlands
Michelangelo Ceci	University of Bari - Italy
Padraig Cunningham	University College Dublin - Ireland
Daniel Gayo-Avello	University of Oviedo - Spain
Ido Guy	IBM Research - USA
Kristian Kersting	University of Bonn - Germany
Matthias Klusch	DFKI GmbH - Germany
Claudia Müller-Birn	FU Berlin - Germany
Alexandre Passant	DERI - Ireland
Claudio Sartori	University of Bologna, Italy
Giovanni Semeraro	University of Bari - Italy
Markus Strohmaier	TU Graz, Austria
Maarten van Someren	University of Amsterdam - The Netherlands
Ugo Vespier	Leiden University, The Netherlands

Invited Talk

Extracting Social and Community Intelligence by Mining Large Scale Digital Footprints

Daqing Zhang, Institute TELECOM SudParis, France

As a result of the recent explosion of sensor-equipped mobile phones, the phenomenal growth of Internet and social network services, the broader use of the Global Positioning System (GPS) in all types of public transportation, and the extensive deployment of sensor network and WiFi in both indoor and outdoor environments, the digital footprints left by people while interacting with cyber-physical spaces are accumulating with an unprecedented speed and scale. The technology trend towards pervasive sensing and computing is making "social and community intelligence (SCI)", a new research area take shape, that aims at mining the "digital footprints" to reveal the human behavior patterns and community dynamics. It is believed that SCI is pushing the research in context-aware computing to a new territory. In this talk, I would like to introduce this emerging research area, present the research background and the evolution of SCI, define the general system framework, discuss the major applications and future research issues of SCI. In particular, I will present our recent work in mining the large scale taxis GPS traces for innovative smart city services and mining the large scale social media data (Foursquare) for community detection and recommendation.

Biography Dr. Daqing Zhang is a professor on Ambient Intelligence and Pervasive System Design at Institute TELECOM SudParis, France. He initiated and led the research in Smart Home, Pervasive elderly care and context-aware computing from 2000 to 2007 at the Institute for Infocomm Research (I2R), Singapore. Dr. Zhang has served as the general chair or program chair for more than 10 international conferences. He is the associate editor for 4 leading journals including ACM Transactions on Intelligent Systems and Technology, Journal of Ambient Intelligence and Humanized Computing (Springer), etc.. He also served in the technical committee for top conferences such as UbiComp, Pervasive, PerCom, etc.

Dr. Zhang is a frequent invited speaker in various international events. His research interests include context-aware computing, pervasive elderly care, sensor based activity recognition, mobile social network, social computing, etc.. He has published more than 150 papers in referred journals and conferences, where his work on context model and middleware got more than 3000 citations in last 5 years. In recent years, he has been exploring a new research direction called "social and community intelligence", making use of GPS traces, social media data, web data and mobile sensor data to extract human and community intelligence and enable various services. Daqing Zhang obtained his Ph.D. from University of Rome "La Sapienza" in 1996.

Guiding user groupings

Learning and combining classification for itemset structuring

Mathias Verbeke, Ilija Subašić, and Bettina Berendt
Department of Computer Science
KU Leuven
Belgium
firstname.lastname@cs.kuleuven.be

Abstract. Structuring is one of the fundamental activities needed to understand data. Human structuring activity lies behind many of the datasets found on the Internet that contain grouped instances, such as file or email folders, tags and bookmarks, ontologies and linked data. Understanding the dynamics of large-scale structuring activities is a key prerequisite for theories of individual behaviour in collaborative settings as well as for applications such as recommender systems. In particular, a key question is to what extent the "structurer" – be it human or machine – is driven by his/its own prior structures, and to what extent by the structures created by others such as one's communities.
In this paper, we propose a methodology for identifying these dynamics. The methodology relies on dynamic conceptual clustering, and it simulates an intellectual structuring process operating over an extended period of time. The development of a grouping of dynamically changing items follows a dynamically changing and collectively determined "guiding grouping". The analysis of a real-life dataset of a platform for literature management suggests that even in such a typical "Web 2.0" environment, users are guided somewhat more by their own previous behaviour than by their peers.

Keywords: Collaborative classification, social web mining, user modelling, guided grouping, divergence measure

1 Introduction

Nowadays, people are faced with a huge amount of data, originating from the increasing amount of emails, stored documents, or scientific publications. Just as people order their CD collection according to genre, sort books on shelves by author, or clothes by colour, it lies in the human nature to structure digital items. This is supported by the directory structure on desktop computers or tags for bookmarks, photos, and online publications. The structuring of data is a diverse process, and grouping of a set of items into subsets can be seen as investing it with semantics: a structuring into sets of items, each instantiating a concept.

In addition, these concepts depend on each other: One concept is what the other is not; there is meaning not only in assigning an item to a concept, but also in *not* assigning it to another concept; and the abstract concepts co-evolve with the assignment of concrete instances to them. This feature makes an analysis of

grouping more complex than an analysis just focusing on "bipartite" relations between items and concepts, on the growth of concept vocabulary over time, etc.

Since the end result of grouping will depend on the context, tasks, and previous knowledge of the "structurer", there is no overall "optimal" grouping. Grouping according to needs and knowledge is a prerequisite for the grouping to make sense to the user. Despite the personal nature of this process, people tend to be influenced by others – be it their real-life social circle, their online friends, or social recommender systems, that could steer the user (and the grouping) in a certain direction.

A good understanding of the dynamics of these structuring activities is a key prerequisite for theories of individual behaviour in collaborative settings, and it is necessary to improve the design of current and next generation (social) recommender systems [3, 4]. Furthermore, it can leverage the design of mechanisms that rely on implicit user interactions such as social search [1, 2]. The goal of this paper is to develop a dynamic conceptual clustering [5] that simulates this intellectual structuring process which is able to identify these structuring dynamics. The evolution of the grouping of an individual user is influenced by dynamically changing and collectively determined "guiding grouping(s)", which we will refer to as *guides*. In this paper, we investigated two types of guides. The first one is motivated by a "narrow" view of the structurer's prior grouping behaviour and experience, while the second one starts from a "wider" perspective that also takes peer experience into account in grouping decisions.

The process in which we want to transfer the structuring of one set of items to another set of items is a combination of two data mining tasks. First, we learn a model for this grouping (classification), which can be applied to structure alternative sets of items. We refer to this model as the grouping's *intension*, as opposed to its *extension*, which is the original, un-annotated grouping of the items. The first task starts with an extension of a structuring, and learns its intension as a classification model. The second task is to use the intensions of the peer groupings and apply their classifiers for prediction, to structure a new item. It starts from defining the k nearest peer groupings for a user. To decide on the k nearest peer groupings in a situation where peers group different items, we defined a novel measure of divergence between groupings that may have a different number and identities of items. Once we obtain the k nearest peers, the task is to decide on item-to-group assignment using their groupings. Based on the presence of an item in peer groupings, this decision is based either on the extension of the peer grouping (when a peer already grouped the item) or on its intension (when a peer has not grouped the item yet). By comparing the possible end groupings with the actual end grouping, we see which guide selection is more likely to have determined the actual process.

Our main contributions are: (a) a new data-mining approach that learns an intensional model of user groupings and uses these to group new items. The method can be used to identify structuring dynamics, which simulates an intellectual structuring process operating over an extended period of time, (b) a new

divergence measure to define divergence between groupings of non-identical item sets, and (c) a study of grouping behaviour in a social bookmarking system.

This paper is structured as follows. The following section reviews related work. Section 3 describes the methodology to identify the structuring dynamics via grouping guidance, where we first outline the general system workflow and subsequently present a formalisation of the different parts. Section 4 contains an empirical analysis of this process that centres on item grouping in a real-life data set. Finally, Section 5 concludes the paper and presents an outlook for future work.

2 Related work

Our work is closely related to the research on personalised information organisation, where the goal is to understand how the user can be supported in searching, exploring, and organising digital collections. This field studies how users structure data collections. Current approaches usually concentrate on individual aspects like search (e.g. personalised ranking), rudimentary exploration support and visualisation. In [25] the authors present a taxonomy of the features that can be used for this purpose. They introduce content-based features (e.g., word distributions), content-descriptive features (e.g., keywords or tags describing textual or image content), and context-independent metadata (e.g., creator) features. The key questions from this domain that are relevant to the topic of this paper are: How is it possible to support a user in grouping collections of items?, How can we use information about the way a user groups collections to support him in structuring as-yet-unknown collections? Nürnberger and Stober [18] give an answer to this question. They describe an adaptive clustering approach that adapts to the user's way of structuring, based on a growing self-organising map. As illustrated in [19], hierarchical clustering can help to structure unseen collections based on the way a user structures her own collections. It models the structuring criteria of a user by constraints in order to learn feature weights. The goal of our approach is to examine these structuring dynamics and investigate how a user is guided by himself or his peers. This can be very useful in detecting a user's guiding preferences and obtaining an overview of the user characteristics in the overall system.

In characterisation studies, system usage patterns are studied to propose models which explain and predict user behaviour in these systems. A number of studies have investigated tagging systems. Closest to our work is the research by Santos-Neto et al. [6] on individual and social behaviour in tagging systems. They define interest-sharing between users as a measure for inferring an implicit social structure for the tagging community. In addition, they investigate whether a high level of tag reuse results in users that tag overlapping sets of items and/or use overlapping sets of tags. Sen et al. [10] study how a tagging community's vocabulary of tags forms the basis for social navigation and shared expression. They present a user-centric model of vocabulary evolution in tagging communities based on community influence and personal tendency, with an evaluation on a real system, namely the *MovieLens* recommender system. Musto et al. [17]

investigate the relation between tagging based on one's own past tags or on the community's tags, abstracting from the dynamics of tag assignments.

However, the studies of tagging are limited to a "bipartite" view of the binary relations between tags and items (or the ternary ones between tags, items and users). They do not take into account the relations between tag assignments and tag non-assignments, i.e. the way in which different concepts interact structurally, for example by being in competition with each other.

Our methodology complements these results with insights into structuring dynamics and behavioural characteristics in collaborative settings. We focus on the way users group items and how they are influenced during this process. This can help to improve the design of future collaborative systems and leverage the design of mechanisms that rely on implicit user interactions.

From conceptual and predictive clustering [7, 8], we used the key idea to a) form clusters of elements, then b) learn classifiers that reconstruct these clusters and c) apply the classifier for further result sets. Research on ontology re-use, in particular adaptive ontology re-use [9], investigates the modelling and mapping steps needed for re-using given ontologies, for whatever purpose. In contrast, we concentrate on re-use for grouping/classifying new objects, and on how to find the most suitable ontologies for re-use.

3 Grouping guidance

This section introduces notation (Section 3.1) and describes the system's general workflow and the individual steps.

The system observes, for each user, an initial grouping and learns an initial classifier from it (Section 3.2). It then identifies, for a given user, which classifier to use next – the user's own or one determined by his peers (Section 3.3). It applies this classifier (Section 3.4) and then updates it to reflect the new item grouping, which now also contains the just-added item (Section 3.5).

Steps 2, 3 and 4 are iterated on any item that the users want to structure. To analyse the structuring behaviour of each user, we compare the resulting groupings of this process to the real user groupings, which are the result of the user' structuring without computational assistance.

In Section 3.3, we present our measure of divergence between user groupings of non-identical item sets.

3.1 Notation

Basic components We use the following notation:

- Let U denote the set of all users (used symbols: u, v, w).
- Let T denote the set of all time points $\{0, 1, ..., tmax\}$, where $tmax$ represents the time at which the last item arrives.
- Let D denote the set of all items (used symbol: d).

D will be used in combination with subscripts that represent users. Its superscripts represent time points. Thus $D_u^t \subseteq D$ denotes the set of all $d \in D$ already considered by $u \in U$ at $t \in T$. The item assigned to the structure by user u at t is an addition to this; it is represented by $d_u^t \in (D \setminus D_u^t)$.

Groupings and classifiers G and C are the (machine-induced) groupings for each user's items and the classifiers learned from them, respectively. G can be any of following:

OG **- Observed Grouping** This represents the grouping of the user at the start of the learning phase. As will be clarified in subsection 3.2, this will be the grouping used to learn the initial classifier at the start of the cycle.

GS **- Simulated Grouping, guided by self** This represents the grouping, at a certain point in time during the cycle, that is solely generated by the classifier of the user in question. This will be specified in subsection 3.4.

Gn **- Simulated Grouping, guided by** n **peers** Just as GS, this represents the grouping at a certain point in time during the cycle, but now the grouping is guided by n peers of the user under consideration. This will be specified in subsection 3.4.

As was the case for item sets, subscripts denote users and superscripts denote time points. G_u^t is the grouping that holds at time t for user u (the set of u's item sets), i.e. a function that partitions the user's current items into groups: $G_u^t : D_u^t \mapsto 2^D$. We will refer to this as the *extensional definition* of a concept, where the definition is formed by listing all objects that fall under that definition, and thus belong to the set or class for which the definition needs to be formed.

C_u^t is the classifier (intension) learned from G_u^t, modelled as the function specifying an item's group according to this classifier: $C_u^t(d) = x$ with $C_u^t : D \mapsto 2^D$. The intensional definition is a concept that originated from logic and mathematics. In this context, an *intensional definition* is defined as a set of conditions for an object to belong to a certain class. This is done by specifying all the common properties of the objects that are part of this set, with the goal of capturing its meaning. Analogously to OG, GS and Gn for groupings, OC, CS and Cn are defined as the initial classifier and the classifiers guided by "self" and by n peers, respectively.

Time Structuring is modelled as evolving through a series of discrete time points, where each new structuring-related activity (such as tagging a document in a social-bookmarking system) happens at some time point and leads to either the assimilation of the item into the given conceptual structure of G and C, or to a modification of the conceptual structure to accommodate the item. The conceptual structure remains stable until the subsequent activity.

We will refer to $next(t, u)$ with $next : T \times U \mapsto T$ as the first time point after t at which u structures an item, which changes the item set to $D_u^{next(t,u)} = D_u^t \cup \{d_u^t\}$. For each time point t' between subsequent considerations of items, grouping and classification remain unchanged. Formally, $\forall t' = t+1, ..., next(t, u) : D_u^{t'} = D_u^{next(t,u)} \wedge C_u^{t'} = C_u^{next(t,u)} \wedge G_u^{t'} = G_u^{next(t,u)}$.

3.2 Initial classifier learning

The task is to classify an item according to a set of concepts, based on its contents. We regard a classifier as an explanation why an object is assigned to a

certain concept, i.e. its intensional definition. As indicated above, the opposite approach is the extensional definition, obtained by listing all objects that fall under that definition, and thus belong to the set or class for which the definition needs to be formed.

The goal is to determine intensional definitions for the user-generated groupings. Each cluster or group is then regarded as a class for which a definition needs to be calculated. Since different algorithms can be used for clustering, there are different ways in which these definitions can be calculated. These intensional definitions can then be used to assign new items to these clusters or groups. This can be seen as a classification task, since new, unclassified items need to be categorised.

3.3 Identifying which classifier(s) to use

The selection of peer guides whose classifiers are used in grouping requires a measure of user similarity or divergence. Each user organises a set of items which he is interested in. Multiple peer users organise different, possibly overlapping, subsets of D. We start from the idea that similarity of items in peers' groupings indicates their interest similarity. The question that arises is: how to define similarity/divergence between groupings of non-identical item sets? Using a measure based on the overlap of items, such as Jaccard index, or mutual information would fit as the similarity/divergence measure if item sets overlap to a large extent. However, it is usually the case that peers have access (or interest) to a limited number of items, and only a small number of popular (easily accessed) items are available to all peers. To overcome this, we define a new measure of divergence between peers.

We assume that a user groups items based on their features, e.g. text for documents. For users u and v and their respective groupings G_u^t and G_v^t, we define the inter-guide measure of diversity *udiv* as:

$$\begin{aligned} udiv(u,v) = \frac{1}{2}(\; & 1/|G_u^t| \sum_{x \in G_u^t} min_{y \in G_v^t} gdiv(x,y) \\ & +1/|G_v^t| \sum_{y \in G_v^t} min_{x \in G_u^t} gdiv(y,x) \;) \end{aligned} \tag{1}$$

where *gdiv(x,y)*, or inter-group diversity, is any divergence measure defined on two groups.[1]

The measure defined in Eq. 1 captures the differences between groupings by rewarding groups having shared items and groups containing similar items. The double average creates a symmetric measure.

We calculate the inter-guide divergence for all pairs of users $u, v \in U$ and create a $|U| \times |U|$ matrix M_{sim}. The M_{sim} matrix is used to select the most similar users to the user who is structuring items. For each user, we extract the

[1] If desired, a similarity measure can be calculated from this easily by appropriate normalisation: $usim(u,v) = (max_{w,z \in U} udiv(w,z) - udiv(u,v))/(max_{w,z \in U} udiv(w,z)$.

corresponding row from M_{sim} and sort it to find the most similar peers, defined as the least divergent ones. From this list, we select the top n users as guide(s).

3.4 Classification

In the classification step, the identified determining classifiers of the peer users from the previous step are now used to classify the item under consideration. We distinguish two cases: in the first one, the user guides himself, so the intensional model of his own structuring is used as classifier for the new item. In the other case, the user is guided by his peers.

Self-guided classification If the user has not seen the item yet, we can use the intensional description of the user's current clustering to classify the new[2] instance.

At time t, the time where the new item arrives, the user's own classifier is applied: $x = CS_u^t(d)$. This gives the proposed group x based on the intensional model of the user under consideration. The item d is added to this x, which results in the new *grouping* $GS_u^{next(t,u)}$.

This is illustrated in Figure 1, which shows the self-guided classification of a user u. It starts from the observed grouping at time point 0, OG_u^0, which is learned via classifier learning from the extensional definitions, i.e. the original grouping of user u. At each time step, when an item arrives, the previous intensional definition, i.e. classifier, from user u is used to classify the new item. After the item is added, a new intensional definition is learned. For example, when item b arrives, CS_u^1 is used to classify it, and a new classifier CS_u^2 is learned.

Peer-guided classification An alternative to self-guided grouping is to use the intensional descriptions of peer users to classify the new item. We distinguish two cases: guidance by the single most similar user, and guidance by the top k peers.

Top-1 peer If the peer user has already seen the item d, she has grouped it into one of her own groups; call this y. If she has not seen the item yet, she would classify it into her present structure into some $C1_u^t(d)$; call this y too. (Recall that the classifier is named $C1_u^t$ because the peer is u's top-1 peer at the current time.) This structuring is now projected back onto u: he will group this item into that of his own groups that is *most similar to* y, or *least dissimilar from* y, namely into his group $x = argmin_{x \in G1_u^t} gdiv(x, y)$. In both cases, d is added to the resulting x. The new grouping is represented by $G1_u^{next(t,u)}$.

This is illustrated in Figure 2, which shows the top-1 peer-guided classification of a user v, with peers u and w. It starts from the observed grouping at time point 0, OG_v^0, which is learned via classifier learning from the extensional

[2] Note that there is another possible case, namely when an item arrives that a user has already seen. However, we were able to disregard this case given the nature of our data. We used *CiteULike* data, which is a social bookmarking system for scientific papers, where we determined the groupings based on the taggings. As already shown in [6], only a very small proportion of the taggings are produced by the user who originally introduced the item to the system, and in general, users do not add new tags to describe the items they collected and annotated once.

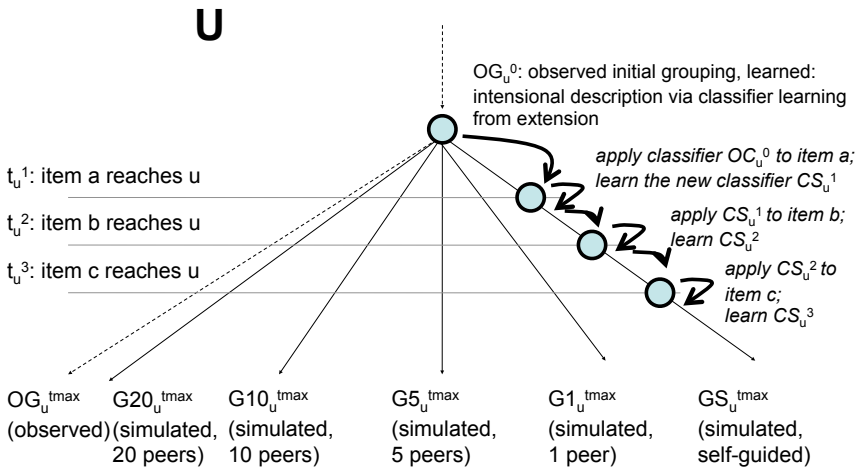


Fig. 1. General workflow for self-guided classification.

definitions, i.e. the original groupings of user v. At each time step, when an item arrives, the most similar user is determined, and her intensional definition is used to classify the new item. After the item is added, a new intensional definition is learned. E.g. at the arrival of item h, user u is identified as the most similar peer of user v, so consequently CS_u^1 is used to classify item h. After the item is added, a new classifier CS_v^2 is learned.

More than 1 peer Let $v_1, ..., v_k$ be the top peers of u at t. For each v, a clustering of user u's grouping is calculated, with the method described for the top-1 peer. Then $y = CS_v^t(d)$ is some $y \in GS_v^t$ (v's group into which the item would be put). The result is a set of groupings, where majority voting decides on the final user group to put the item in. In case of ties, the largest $x \in Gn_u^t$ is chosen.

3.5 Intension update

After the addition of the newly classified item, the intensional definition needs to be updated, which is needed to reflect the changed grouping structure implied by this addition. This is done by updating the classifier, which results in the new intensional definitions $CS_u^{next(t,u)}$, $C1_u^{next(t,u)}$ and $Cn_u^{next(t,u)}$ for the self-guided, top-1 and top-n peer-guided classification respectively.

4 Empirical results

For testing the method outlined in the previous sections we ran a series of empirical analyses on *CiteULike*[3], a social bookmarking system designed for scientific publications. A social bookmarking system is a perfect candidate for our grouping approach. Its Web 2.0 nature implies collaboration between peer users for enhancing the system, and by tagging, users implicitly group items. This was further motivated by the results of [6], which indicate that a rather low level of item re-tagging and tag reuse, together with the much larger number of

[3] http://www.citeulike.org

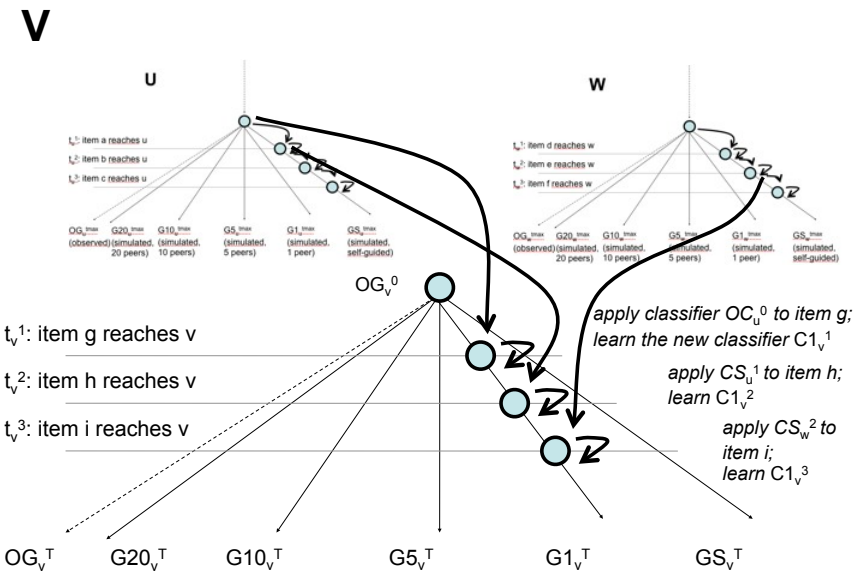


Fig. 2. General workflow for top-1 peer-guided classification.

items than tags in CiteULike, suggests that users exploit tags as an instrument to categorise items according to topics of interest. This is also indicated by results on usage patterns of collaborative tagging systems in [16]. Furthermore, they also indicate that the relatively high level of tag reuse suggests that users may have common interest over some topics. This motivated us to test whether this collaborative power and shared interests are the basis for users' organisations of documents, or whether they rather "trust" their own experience and are not "guided" by their peers.

In a social bookmarking system users apply free text keywords (tags) to describe an item. One application of a tag by a user to a item is referred to as *tagging*. Combining multiple taggings by multiple users gives rise to a data structure that has been termed *folksonomy* [15]. A folksonomy can be viewed as a tripartite graph where nodes belong to sets of users, tags, or items and a hyper-edge is created between between a user, tag, and item for each tagging.

4.1 Dataset

We obtained a dump of the CiteULike database containing all entries from December 2004 to February 2010. In total, there were 486,250 unique tags, 70,230 users, 2,356,013 documents, and 10,236,568 taggings. As noted by previous work [16], the nodes and hyper-edges in a folksonomy have a long tail distribution, and most users tag a small number of items, and most items are tagged rarely. To overcome this sparsity, we followed a widely used folksonomy sampling method based on p-core subgraphs. p-core subgraphs of a graph are its connected components in which the degree of each node must be at least equal to p. As in similar studies [13, 14], we set the value for p to 5. Further, we constrained the dataset with regard to time and analysed only the taggings from 01/2009 until

02/2010. In total this dataset had 12,982 tags, 377 users, 11,400 documents, and 124,976 taggings. We refer to this folksonomy as F. For each document in the p-cores, we obtained the text of the abstract available on the CiteULike website.

4.2 Initial grouping

There are structures where people group explicitly, such as mail folders, and others where people group implicitly. Tagging is an example of the latter. Since organising and structuring items is one of the largest incentives for using a social bookmarking system [10], we assume that the tagging assignments the user has at the start of the learning phase are a reasonable starting point to create the initial groupings. In accordance with Section 3.1, we use the word "observed" for these groupings, fully aware that these are generated by clustering, but we wanted to make the distinction based on the data which was used for grouping (observed data).

In order to learn those observed groupings, we split the dataset into two parts. The first part, containing the first 7 months of our dataset, is used for learning the initial groupings $G^0_\bullet$. On this part, we apply a clustering algorithm as in [23, 19]. For a user u, we first restrict F to contain only nodes connected to u. This gives rise to a graph F^u. We then project F^u to a graph SF^u in which nodes belong to documents from set D^0_u and edges are created between these nodes when some tag is applied to both documents. The weight of the edges in SF^u is equal to the number of tags that two documents share in F^u. We then applied a modularity clustering algorithm [26] to partition the SF^u graph. Each partition is treated as one group in G^0_u. This is repeated for all $u \in U$ to obtain initial groupings of all users.

The next step is to learn the initial classifiers. For this purpose, we used the Naive Bayes Classifier implementation in the WEKA data mining toolkit [21]. We used the bag of words representation of each publication's abstract as features for classification. This motivates our choice for Naive Bayes, since it has the property of being particularly suited when the dimensionality of the inputs is high. We also used a kernel estimator for modelling the attributes, rather than a single normal distribution, since this resulted in a better classification accuracy. The kernel estimator uses one Gaussian per observed value in the training data, and the conditional probability output by a kernel estimator is the weighted combination of the conditional probabilities computed from each of its constituent Gaussians.

4.3 Simulating groupings

We represent groups belonging to a grouping using language models. At time t for a user u for every of his groups $x \in G^t_u$ we create a language model Θ_x. To find the most similar peers to a user u we calculate his inter-guide divergence (Eq. 1) to all users $v \in U$. In our experiment on social bookmarking, as the inter-group divergence (*gdiv*) we used Jensen-Shannon divergence (JS) [27]. For two language models Θ_x and Θ_y representing groups x and y belonging to groupings $G^\bullet_u$ and $G^\bullet_v$, JS is defined as:

$$JS(\Theta_x, \Theta_y) = \frac{1}{2}KL(\Theta_x, \Theta_z) + \frac{1}{2}KL(\Theta_y, \Theta_z), \quad (2)$$

where the probability of every word in Θ_z is its the average probability in Θ_x and Θ_y; $KL(\Theta_\bullet, \Theta_*)$ is Kullback-Leibler divergence between two language models.

4.4 Results

Once we obtained the groupings for GS and $G\{1|5|10|20\}$ of one user, we compared these groupings with his observed groupings at $tmax$. Like the observed initial grouping of a user u, his observed final grouping at $tmax$ is a structuring of all the documents he has considered at that time. Every simulation run (whether it be guided by self, the top-1 peer, or the top-k peers) also considers the same sequence of documents arriving for u. Therefore, all simulated groupings and the observed grouping of one user at $tmax$ contain the same document set. To compare these groupings, we investigated the similarity between them. Since the groupings to compare contain the same set of documents, we can use *normalised mutual information* (NMI). It has desirable properties: it has a value between 0 and 1, and it behaves well when calculated for groupings with different numbers of groups [22].

The normalised mutual information (specifically, NMI 4 [20]) of two groupings G and G' is defined as

$$NMI(G, G') = H(G) + H(G') - \frac{H(G, G')}{\sqrt{H(G)H(G')}} \quad (3)$$

where $H(G)$ is the entropy of grouping G and $H(G, G')$ the joint entropy of G and G' together.[4]

The similarity results for all groupings are shown in Figure 3. Our research question was to investigate whether users are guided in structuring items by their own experience or by their peers. The results suggest that in spite of the collaborative nature of a social bookmarking system, users tend to keep to their structuring system. The NMI between OG and GS is highest having a mean of 0.61 (st.dev: 0.2). Using the closest peer grouping for grouping ($G1$) does not produce more similar groupings, having a mean of 0.57 (st.dev: 0.21). Including more guides into decision does not improve the results; in our experiment the best peer grouping is for $G20$ (mean: 0.58, st.dev: 0.21). This is more similar to OS when compared to $G1$, but still less similar compared to GS.

Given the high standard deviation for all groupings, we investigated whether some users are more influenced by some guides. To discover this, we looked more closely into the distribution of NMI across all users. Figure 4 shows the results. We expected to see a more bimodal distribution if some users "prefer" different guides. However, all distributions fit to a normal distribution skewed to the right (towards higher similarity). Normality of the distribution is tested using

[4] for groups x, y and p the distribution of items over them:
$H(G) = -\sum_{x \in G} p(x) log_2 p(x)$ and $H(G, G') = -\sum_{x \in G} \sum_{y \in G'} p(x, y) log_2\, p(x, y)$.

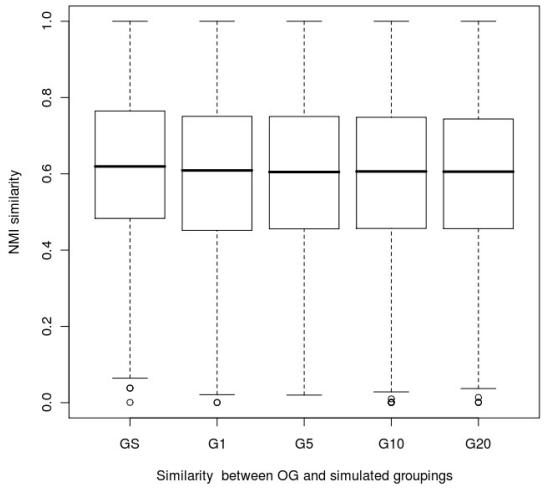


Fig. 3. Similarity between OG and simulated groupings.

the Shapiro-Wilk test, and all p values are higher than 0.05. This suggests that the majority of users behave in the same manner when it comes to structuring their items.

The results should be interpreted with some caution, and they lead to further research questions. The differences of fit were not very large and not significant. And a result that shows that on average, a user′s observed grouping is 0.61 similar to his simulated self-guided result and 0.57 to the simulated peer-guided result, can mean two things: the simulated self-guided result and peer-guided result are quite similar, or they are quite different. We will explore this question further by considering our results in the light of related work.

Relation to other research. The results correspond to the results by Santos-Neto et al. [6] on individual and social behaviour in tagging system. The authors used a solution based on interest-sharing between users to infer an implicit social structure for the tagging community. They defined the interest-sharing graph, where users are connected if they have tagged the same item or used the same tags and found out that users self-organise in three distinct regions: users with low activity and unique preferences for items and tags, users with high similarity among them, but isolated from the rest of the system, and a large number of users with mixed levels of interest sharing. The results are also compatible with the findings of [16], which – based on the usage patterns in collaborative tagging systems – indicates that users are drawn primarily to tagging systems by their personal content management needs, as opposed to the desire to collaborate with others. Our results extend those findings: Our approach not only looks at the tags they use, but also at the way they use the tags vis-à-vis one another, i.e. in their structuring. This can give an indication as to whether users are more self-guided than peer-guided.

The results of [17] may at first sight appear to point into a different direction: Musto et al. studied the accuracy of tag prediction based on self-guidance and

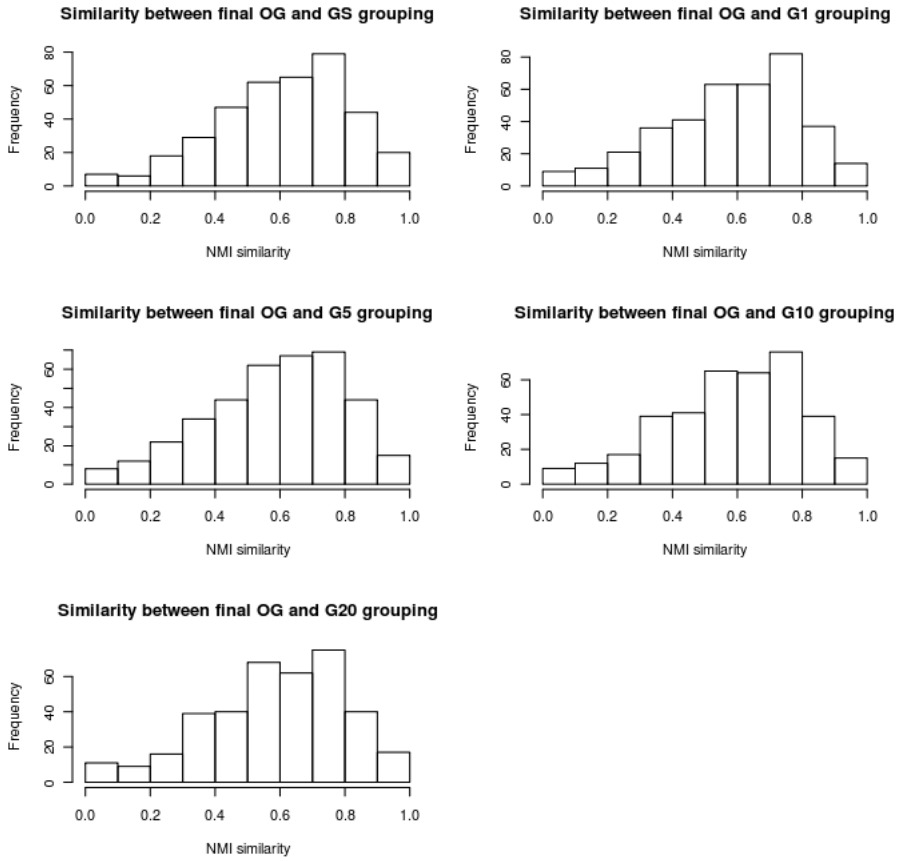


Fig. 4. Distribution of similarity between OG and simulated groupings.

community-guidance (in our terminology: all peers). They found that (a) the community-guided tag predictions were somewhat more accurate than the self-guided ones and that (b) a strategy that first extracts tag suggestions from the items themselves and then complements them with personal tags and then the community's tags produced the best accuracy. Closer inspection reveals, however, that the basic assumptions of that model and of our own form an interesting complementarity and may help to further interpret our results.

The basic difference is that we are not interested in the tags per se, but in the mental structure that they reveal. We will explain the complementarity using a very simple fictitious example. Assume a user u and a user v (who represents the whole community). u has tagged items D_1 (some set of items) with the tag tag_1, and D_2 with tag_2. v has tagged some D_3 with tag_3 and some D_4 with tag_4. All 4 tags are different. A new item arrives for u that is most similar to his group D_1 and to the community's group D_3. If the user is guided by self, our model will predict the new item to go into D_1; if he is peer-guided, our model will predict that the new item goes *into that of u's groups that is most similar to* D_3. Now assume that D_1 is most similar to D_3. This corresponds to a situation

in which everybody more or less structures in the same way. Our model will then predict that the new item goes into D_1. And the final observed grouping will be quite similar to the self-guided simulated one, and also to the peer-guided simulated one. (And this is what we found.) Musto et al., in contrast, look at the tag manifestations of this mental structure. Their model predicts for self-guidance that the new item will be tagged with tag_1, and for peer-guidance that it will be tagged with tag_3. So if vocabulary converges in the population, the item will be tagged with tag_1 *and* tag_3, and the personal-plus-community model will produce the best prediction. (And this is what Musto et al. found.) Thus, one interpretation that is consistent with the results of both studies is that while there is some more self-guidance, the differences to peers are actually quite small, and individual vocabularies do tend to converge to the community's vocabulary.

This interpretation however relies on an assumption about dynamics (the convergence of vocabulary), and dynamics were not modelled by Musto et al. Also, there are combinations of equality/difference in grouping on the one hand and vocabulary on the other, that are associated with different predictions in the two models. These combinations may persist in specific subcommunities. Finally, the levels of accuracy/NMI that they/we measured indicate that other factors play an important role too in tagging. In future work, this should be investigated in more detail.

5 Conclusions and outlook

Our high level interest during this research was to investigate collaborative grouping of items and to build a framework that simulates this process. Specifically, we were interested in how different users structure items depending on the influence that guides this structuring process. We developed a new methodology that learns and combines classifiers for item set structuring. This methodology starts by, in the first step, learning a model for an existing grouping, which we referred to as the intensional definition. The second step uses these learned intensions to classify new items. To decide on the most appropriate users to take into account when grouping a new item, we defined a new divergence measure between groupings that may have different numbers and identities of elements. This methodology is applied to simulate the intellectual structuring process which underlies these structuring dynamics. We tested this approach on *CiteULike*, a social-bookmarking platform for literature management. The results of the study can have implications for system design of recommender systems and social search methods, for which a good understanding of the dynamics of these grouping activities is a prerequisite.

Limitations. The main question we addressed was one of the difference in groupings, and we did not look at the benefits users have from adopting some guides. As a baseline we used *observed groupings*, which are not explicit user groupings, but implicit groupings learned based on users tag assignments. We are aware of the possible bias here and do not claim that these are the "best" groupings users can have. Also, we used a rather simple classifier, and a limited dataset. However, the method we built in order to combine groupings can easily

be extended. Our analysis provides insights both into the grouping behaviour in general and into the behaviour of users in social bookmarking systems.

Future work. The work presented in this paper complements our previous work, where we presented a diversity-aware method for sense-making in document search [24]. However, many open issues remain to be solved. First of all, this methodology can be of use in different applications for (tag) recommendation and social search, where the grouping dynamics and behaviour adds a new level to the current individual and social measures used by these systems. Furthermore, we could extend the methodology to regrouping own groupings, based on the groupings of one's peers. The proposed method can also be combined with other metrics to create a hybrid measure for item set structuring.

Conclusion. We presented a study into grouping behaviour of users. Our framework combines different data mining methods to simulate collaborative grouping of items. The results of our experiment suggest that even in such open systems as social bookmarking tools, people tend to "trust" their own experience more than turn to the wisdom of the crowd. The main question we wish to follow in the future is not one of trust, but one of "benefit" users get by being able to choose from a myriad of diverse groupings of the items they are interested in.

References

1. Evans, B. M., Chi, E. H.: Towards a model of understanding social search. In: Proc. of the 2008 ACM conference on Computer supported cooperative work, pp. 485–494. ACM, San Diego, CA, USA (2008)
2. Carmel, D., Zwerdling, N., Guy, I., Ofek-Koifman, S., Har'el, N., Ronen, I., Uziel, E., Yogev, S., Chernov, S.: Personalized social search based on the user's social network. In: Proceedings of the 18th ACM Conference on Information and Knowledge Management, 1227–1236 (2009)
3. Adomavicius, G., Tuzhilin A.: Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. In: IEEE Transactions on Knowledge and Data Engineering, 734–749. IEEE Computer Society, Los Alamitos, CA, USA (2005)
4. Herlocker, J. L., Konstan, J. A., Terveen, L. G., Riedl, J. T.: Evaluating collaborative filtering recommender systems. In: ACM Transactions on Information Systems, 5–53. ACM, New York, NY, USA (2004)
5. Michalski, R. S., Stepp, R. E.: Learning From Observation: Conceptual Clustering. In: Machine Learning: An Artificial Intelligence Approach, 331–364 (1983)
6. Santos-Neto, E., Condon, D., Andrade, N., Iamnitchi, A., Ripeanu, M.: Individual and social behavior in tagging systems. In: Proceedings of the 20th ACM conference on Hypertext and hypermedia, 183–192. ACM, New York, NY, USA (2009)
7. Blockeel, H., De Raedt, L., Ramon, J.: Top-Down Induction of Clustering Trees. In: Proceedings of the 15th International Conference on Machine Learning, 55–63. Morgan Kaufmann (1998)
8. Zenko, B., Džeroski, S., Struyf, J.: Learning Predictive Clustering Rules. In: Proceedings of the 4th International Workshop on Knowledge Discovery in Inductive Databases: Revised Selected and Invited Papers, 234–250. Springer (2005)
9. Stecher, R., Niederée, C., Nejdl, W., Bouquet, P.: Adaptive ontology re-use: finding and re-using sub-ontologies. IJWIS. 4 (2), 198–214 (2008)
10. Sen, S., Lam, S. K., Rashid, A. M., Cosley, D., Frankowski, D., Osterhouse, J., Harper, F. M., Riedl, J.: tagging, communities, vocabulary, evolution. In: Proceed-

ings of the 2006 20th anniversary conference on Computer supported cooperative work, 181–190. ACM, New York, NY, USA (2006)
11. Wang, J., Clements, M., Yang, J., de Vries, A. P., Reinders, M.J.T.: Personalized Collaborative Tagging. In: Information Processing & Management (2009)
12. Heymann, P., Garcia-Molina, H.: Collaborative Creation of Communal Hierarchical Taxonomies in Social Tagging Systems. Technical Report, Technical Report (2006)
13. Schmitz, C., Hotho, A., Jäschke, R., Stumme, G.: Mining association rules in folksonomies. In: Data Science and Classification, pp. 261–270. Springer (2006).
14. Jäschke, R., Marinho, L., Hotho, A., Schmidt-Thieme, L., Stumme, G.: Tag Recommendations in Folksonomies. In: Proceedings of the Lernen - Wissensentdeckung - Adaptivität (2007)
15. Mika, P.: Ontologies are us: A unified model of social networks and semantics. In: Proceedings of the International Semantic Web Conference, pp. 522–536 (2005)
16. Golder, S. A., Huberman, B. A.: Usage patterns of collaborative tagging systems. Journal of Information Science. 32 (2), 198–208 (2006)
17. Musto, C., Narducci, F., Lops, P., de Gemmis, M.: Combining Collaborative and Content-Based Techniques for Tag Recommendation. In: Proceedings of the 11th International Conference on E-Commerce and Web Technologies (EC-Web 2010), 13–23. Springer (2010)
18. Nürnberger, A., Stober, S.: User Modelling for Interactive User-Adaptive Collection Structuring. In: Adaptive Multimedial Retrieval: Retrieval, User, and Semantics, pp. 95–108. Springer-Verlag, Berlin, Heidelberg (2008).
19. Bade, K., Nürnberger, A.: Creating a Cluster Hierarchy under Constraints of a Partially Known Hierarchy. In: Proceedings of the 2008 International Conference on Data Mining, pp. 13–24 (2008)
20. Strehl, A., Ghosh, J.: Cluster ensembles - a knowledge reuse framework for combining multiple partitions. Journal of Machine Learning Research. 3, 583–617 (2003)
21. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I. H.: The WEKA data mining software: an update. SIGKDD Explor. Newsl. 11 (1), 10–18 (2009)
22. Pfitzner, D., Leibbrandt, R., Powers, D.: Characterization and evaluation of similarity measures for pairs of clusterings. Knowledge and Information Systems. 19 (3), 361–394 (2009)
23. Begelman, G., Keller, P., Smadja, F.: Automated Tag Clustering: Improving search and exploration in the tag space. In: Proceedings of the Collaborative Web Tagging Workshop at WWW06 (2006)
24. Verbeke, M., Berendt, B., Nijssen, S.: Data mining, interactive semantic structuring, and collaboration: a diversity-aware method for sense-making in search. In: Proceedings of First International Workshop on Living Web, Collocated with the 8th International Semantic Web Conference (2009)
25. Kashyap, V., Sheth, A.: Semantic Heterogeneity in Global Information Systems: The Role of Metadata, Context and Ontologies. In: Cooperative Information Systems, pp. 139–178, Academic Press, San Diego (1998)
26. Newman, M. E. J., Girvan, M.: Finding and evaluating community structure in networks. In: Physical Review E, American Physical Society. 69 (2) (2004)
27. Briët, J., Harremoës, P.: Properties of classical and quantum Jensen-Shannon divergence. In: Physical Review A, American Physical Society. 79 (5) (2009)

A Fast and Simple Method for Profiling a Population of Twitter Users

Hideki Asoh[1], Kazushi Ikeda[2], and Chihiro Ono[2]

[1] Intelligent Systems Research Institute, AIST
AIST Tsukuba Central 2, 1-1-1 Umezono, Tsukuba, Ibaraki 305-8568 Japan
h.asoh@aist.go.jp
[2] KDDI R&D Laboratories, Inc.
2-1-15 Ohara, Fujimino, Saitama 356-8502 Japan
ono/kz-ikeda@kddilabs.jp

Abstract. Recently, many people have been voicing their opinions via social network services (SNSs) such as Twitter, thus making SNSs powerful real-time marketing tools for investigating users' response to products, events, and contents. In a previous study, we proposed a method for profiling Twitter users on the basis of their tweets and applied this method to various marketing applications. The method estimates the demographics of individual users, such as gender, age, and geographical location. In contrast to applications for which it is important to estimate profiles for individual user – for example, targeted advertising – there are many other marketing applications for which information on the distribution of demographic classes, such as the ratio of males to females, among a population of all responding users is essential. Although such distribution can be calculated from the estimated demographic information of individual users, the estimation process may be accelerated by directly estimating the distribution from the tweets of the responding users. In this paper, we propose a fast and simple method for profiling a population of Twitter users who tweet on an event without needing to estimate the users' individual demographics. In addition, we evaluate the proposed method using real tweet data.

Keywords: Class Ratio Estimation, Population Profiling, Twitter

1 Introduction

Today many people voice their opinions via social network services (SNSs) such as Twitter [8]. which currently has over 140 million users worldwide. With rapidly increasing numbers of users, SNSs are being recognized as powerful real-time marketing tools for investigating users' responses to products, events, and contents [3].

In order to enhance the value of tweets for marketing, it is essential to combine tweets from a user with profile information of the user such as age, gender, occupation, geographical location, and marital status. However, few of Twitter users disclose this information.

In a previous paper [4], we proposed a method for estimating the profiles of individual Twitter users from their tweets. This method use support vector machines (SVMs) [9] to classify a word vector constructed from a user's tweets into a profile for that user. The SVMs are trained with the tweets of users who have disclosed their profile information.

However, in many other marketing applications such as audience rating survey of television programs, it is not the demographic information on individual users that is important, but rather the distribution of demographic classes, such as the ratio of males to females, among a population of the users responding to a specific product, content, or event. Although these ratios can be calculated from the estimated profiles, the process of estimating profiles is time-consuming. Estimating the ratio directly from the tweets of responding users without classifying each user's profile may effectively reduce the computational cost and response time.

This paper proposes a fast and simple method of estimating the distribution of Twitter user profiles without estimating the profiles of individual users, and evaluates the performance of the method using real tweet data.

The rest of this paper is organized as follows. Section 2 presents a brief overview of our previous study of estimating individual profiles on the basis of tweets because some part of the study are used in this study also. Section 3 introduces our proposed method for directly estimating class ratios. Section 4 discusses our experiments using real tweet data. Section 5 describes related studies, and Section 6 presents our conclusion and future research directions.

2 Profile Estimation

In this section, we briefly describe the profile estimation method presented in our previous study [4], in which we constructed SVMs for classifying target users based on their tweets. The input to the SVMs is a word vector extracted from the user's past tweets, and the output is the likelihood that the user belongs to a specific user segment such as "male", "female", "teens", and "twenties", for example, a SVM differentiates "teens" from "not teens".

We automatically collected tweets of users who had disclosed their profile information, and constructed a training dataset that comprised pairs containing a feature word vector and a user segment label. The characteristic words representative of each user segment were selected through statistical feature selection based on Akaike's information criteria [1]. We chose 2,000 characteristic words for each user segment. For example, the characteristic words for the "teens" segment included "school," "classroom," and "examination". A 2,000-dimensional word vector was used to represent whether or not each of the characteristic words appeared in the tweets (1 or 0). Using the data we trained SVMs to differentiate user segments.

In the classification phase, we computed word vectors from the most recent 200 tweets of the target user and input into these into the SVMs. Comparing the output values of the SVMs corresponding to mutually exclusive user segments,

we estimated the profile of the user such as “female”, “twenties”, “student”, “not married”, and “living in Kanto area”.

We evaluated the performance of our proposed profile estimation method using real Twitter data and confirmed its strong performance. For example, the error rate for classification of user into “male” and “not male” was 0.093.

3 Class Ratio Estimation

Estimating the profile of each user is essential for applications such as personalized or targeted advertising. However, in many other marketing applications such as audience measurement for a TV program, what is important is not the demographic information on individual users but the distribution of user segments, such as the ratio of male and female, among responding users to a specific product, content, or event.

Although the distribution can be calculated from the estimated individual profiles of the target users, this estimation process is time consuming. Estimating the distribution directly from the tweets of responding users without individually classifying each user's profile, may effectively reduce the computational cost and response time. In this section, we describe a fast and simple algorithm for class ratio estimation.

3.1 Problem Formulation

The problem of estimating class ratios based on tweets of target users can be formulated as follows. Let $\mathbf{x} \in \{0,1\}^d$ be an input word vector and $y \in \{1, ..., K\}$ be the label of a user segment class. The joint probability distribution of $\mathbf{x}$ and y is

$$p(\mathbf{x}, y) = p(y|\mathbf{x})p(\mathbf{x}) = p(\mathbf{x}|y)p(y).$$

In the usual machine learning setting, the joint probability distributions of the training dataset $p_{train}(\mathbf{x}, y)$ and the test dataset $p_{test}(\mathbf{x}, y)$ are assumed to be identical. In this case, the class ratio $p(y)$ can easily be estimated by merely counting each class label in the training dataset.

However, in the case of class ratio estimation, we can not assume that the joint distributions are the same because the class ratio $p(y)$ is different between training data and test data. Although the class ratio is different, the conditional probability distribution $p(\mathbf{x}|y)$, which represents the tendency of word useage of users in each class, can be considered to be the same for training dataset and test dataset because the users in both datasets come from the same population of all twitter users.

Hence, in the following, we assume that the conditional probability $p(\mathbf{x}|y)$ is the same for both datasets and that the class probability $p(y)$ is different for training and test datasets. That is,

$$p_{train}(\mathbf{x}|y) = p_{test}(\mathbf{x}|y), \tag{1}$$

$$p_{train}(y) \neq p_{test}(y). \tag{2}$$

3.2 Algorithm for Class Ratio Estimation

In order to estimate the class ratio $p_{test}(y)$ for the target users, we focus on $p_{test}(\mathbf{x})$, the distribution of word vectors computed from the target users' tweets. From the definition of joint probability, we have

$$p_{test}(\mathbf{x}) = \sum_y p_{test}(\mathbf{x}, y) = \sum_y p_{test}(\mathbf{x}|y)p_{test}(y).$$

Using assumption (1), we have

$$p_{test}(\mathbf{x}) = \sum_y p_{train}(\mathbf{x}|y)p_{test}(y).$$

Applying the above formula, we can estimate $p_{test}(y)$ by minimizing the distance between $p_{test}(\mathbf{x})$ and $\sum_y p_{train}(\mathbf{x}|y)\theta_y$ for the class ratio parameters $\theta = (\theta_1, ..., \theta_K)$, $\theta_k \geq 0$ and $\sum_k \theta_k = 1$.

Various distance measures between probability distributions can be adopted for the estimation. In this paper, owing to its simplicity, we propose to use the Euclidean distance between the mean vectors of the two distributions, that is,

$$R(\theta) = \left(\int \mathbf{x}\, p_{test}(\mathbf{x}) d\mathbf{x} - \int \mathbf{x} \left(\sum_y p_{train}(\mathbf{x}|y)\theta_y \right) d\mathbf{x} \right)^2.$$

Since

$$\int \mathbf{x} \left(\sum_y p_{train}(\mathbf{x}|y)\theta_y \right) d\mathbf{x} = \sum_y \left(\int \mathbf{x}\, p_{train}(\mathbf{x}|y)\theta_y d\mathbf{x} \right),$$

by introducing the below notations

$$\mu_{test} = \int \mathbf{x}\, p_{test}(\mathbf{x}) d\mathbf{x}$$
$$\mu_{train|y} = \int \mathbf{x}\, p_{train}(\mathbf{x}|y) d\mathbf{x},$$

the criterion to be optimized can be expressed as

$$R(\theta) = \left(\mu_{test} - \sum_y \mu_{train|y}\, \theta_y \right)^2,$$

where μ_{test} is the mean of the word vectors in test dataset and $\mu_{train|y}$ is the class mean of the word vectors in the training dataset.

This is a simple linear optimization problem that can be solved using the generalized inverse matrix of the class mean vectors. Letting M be $(\mu_{train|1}, ..., \mu_{train|K})$,

$$R(\theta) = ||\mu_{test} - M\theta^T||^2$$

where θ^T is the transpose of θ and the optimum $\theta = \theta^*$ can be computed as

$$\theta^* = M^+ \mu_{test},$$

where M^+ is the generalized inverse matrix of M.

In an extreme case when each user segment is degenerated into a mean word vector, this method exactly gives the true class ratios.

4 Experimental Results

We evaluated the effectiveness of our proposed method using real Twitter data. In the experiment, we crawled the tweets of individual users who had disclosed their profile. The target attributes were gender and age. We divided the data into training data and test data. Then using the training data, the mean word vector for each user segment was computed. Using the test data, we prepared several test datasets with different class distributions. We then applied our proposed method to the test data. For comparison, we also applied the method that employs SVM classification. We evaluated the estimation accuracy as the mean square error (MSE) of the ratio vector. That is, where the true class ratio is $(r_1, \cdots, r_K)$ and the estimated ratio is $(\hat{r}_1, \cdots, \hat{r}_K)$, the accuracy was measured as

$$MSE = \sum_{y=1}^{K} (r_k - \hat{r}_k)^2.$$

4.1 Gender Ratio Estimation

For the gender ratio estimation experiment, we collected 200 tweets each from 1,000 users who had disclosed gender information. The characteristic words for the classifications into "male" and "female" identified through the method described in section 2 were combined to form a set of 4000 feature words. Occurrences of these 4,000 gender sensitive words in the collected tweets were examined for making word vectors for individual users. Then each word vector and gender of each user were paired and compiled into the dataset. We randomly divided the dataset into training data and test data. We used 60% for training data and 40% for test data.

Using the test data, we prepared nine test datasets with class ratios ranging from 0.2:1.0 to 1.0:0.2. The gender ratio for the training data was fixed at 1:1. We applied our proposed class ratio estimation method to the test dataset and evaluated the MSE. For comparison, we also performed class ratio estimation using SVM classification. We repeated the entire experiment 10 times and evaluated the mean of the MSE values. The results are shown in Table 1 and Figure 1.

The results demonstrate that the direct method of ratio estimation outperformed the SVM method in many cases. In addition, note that when the ratio of the test dataset was far from that of the training dataset, the performance

Table 1. Accuracy of Gender Ratio Estimation

True Ratio ($female/male$)	Direct Method (MSE)	SVM (MSE)
0.2	0.0837	0.157
0.4	0.0621	0.112
0.6	0.0458	0.077
0.8	0.0354	0.048
1.0	0.0329	0.0295
1.25	0.0315	0.0153
1.67	0.0278	0.0234
2.5	0.0334	0.0513
5.0	0.0436	0.0931

Table 2. Accuracy of Age Ratio Estimation

True Ratio ($30s/40s$)	Direct Method (MSE)	SVM (MSE)
0.25	0.164	0.141
0.5	0.121	0.096
1.0	0.074	0.049
2.0	0.079	0.042
4.0	0.115	0.066

of our proposed method degenerate slower than the SVM method. We used the statistical computing language R for our experiments and found that the direct method was 62.5 times faster than the method using SVMs.

4.2 Age Ratio Estimation

The problem of gender ratio estimation is a two class problem ("male" and "female"). In order to evaluate the effectiveness of our method for multi-class problems, we conducted a second experiment that estimated the ratio of age classes. In the experiment, we collected 200 tweets each from 756 users who had disclosed their age information. Age was divided into 4 classes (teens:10-19, twenties:20-29, thirties:30-39, and over 40). We combined the feature words for the four classes to form a set of 8,000 feature words. As in the previous experiment, we divided the data into training data and test data. On the basis of the test data, we prepared five test datasets with "twenties":"thirties" class ratios ranging from 0.25:1.0 to 1.0:0.25. The ratio of the other age classes were fixed to constant. The ratio among the training data was also fixed at 1:1:1:1. The results are shown in Table 2 and Figure 2. The results show that the performance of the SVM method was superior to that of the direct method in this case.

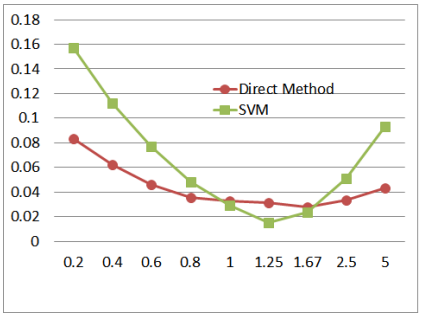


Fig. 1. MSE for Gender Ratio Estimation

Fig. 2. MSE for Age Ratio Estimation

5 Related Work

The assumptions (1) and (2) look similar to the "covariate shift" [7] where

$$p_{train}(y|\mathbf{x}) = p_{test}(y|\mathbf{x}),$$
$$p_{train}(\mathbf{x}) \neq p_{test}(\mathbf{x})$$

are assumed. Different from the covariate shift, in our case we can not assume $p_{train}(y|\mathbf{x}) = p_{test}(y|\mathbf{x})$. This means that the classifier which is trained by the training dataset can be biased for the test dataset when the class ratio of the training dataset and the test data set are different. This may be a reason for that the SVM method did not work well in our gender ratio estimation experiment when the ratio of the test dataset is far from that of the training dataset. Figure 1 shows a kind of bias-variance trade-off [2].

Saerence et al. treated the problem and proposed a method for correcting the bias by estimating class ratio based on the expectation-maximization (EM) algorithm and demonstrated that the performance of classification is improved [6]. However the iterative method is computationally heavy and difficult to apply to our problem.

As for the problem of directly estimating class ratios, recently, Du Plessis et al. reformulated the problem as distribution matching and proposed algorithms that minimize Kullback-Leibler (KL) divergence and Pearson divergence [5]. However, their algorithms use nonparametric approximation of the divergence using Gaussian kernel function and are, therefore, difficult to apply to our high dimensional binary word vector data.

The main contributions of the present study are spotlighting the class ratio estimation problem in the context of social media analysis for marketing applications, proposing a fast and simple method that can be applied to large-scale and high-dimensional SNS data, and demonstrating the effectiveness of the proposed method.

6 Conclusion and Future Work

In this paper, we proposed a fast and simple method for class ratio estimation and applied it to profiling a population of Twitter users. The experiments with real tweet data demonstrate that our proposed method outperforms the method using SVM in the case of a two class problems and is therefore promising.

Future work includes more intensive evaluation of our proposed method over various datasets for different class ratio estimation problems. In order to improve the accuracy of the class ratio estimation, a promising way to extend the present study may be to take into consideration of information on $p_{test}(\mathbf{x})$ that is more detailed than the simple mean vector. Following the idea of the KL-divergence minimization in [5] may be promising. From the view point of bias-variance trade-off, extending the evaluations presented in section 4 using low variance classifiers, such as logistic regression, decision tree, or naive Bayes, may be interesting. Theoretical analysis about the performance of the proposed method may be feasible and will become another clue to improve the performance. It would also be interesting to apply our method to other SNSs such as blogs.

Acknowledgments. We thank anonymous reviewers for their helpful comments. We are grateful to Dr. Yasuki Nakajima, president of KDDI R&D Laboratories, for his continuous support for this research.

References

1. Akaike, H.: A new look at the statistical model identification, *IEEE Transactions on Automatic Control*, 19(6), 716–723 (1974)
2. Duda, R. O., Hart, P. E., and Stork, D. G.: *Pattern Classification*, Wiley-Interscience (2000)
3. Esparza, S. G., O'Mahony M. P., and Smyth, B.: Real-time web as a source of recommendation knowledge, *Proceedings of the 4th ACM Conference on Recommender Systems* (2010)
4. Ikeda, K., Hattori, G., Ono, C., and Takishima, Y.: Social media visualization for TV, *Proceedings of the International Broadcasting Convention (IBC 2011) Conference*, D-243 (2011)
5. Du Plessis, M. C. and Sugiyama, M.: Semi-supervised learning of class balance under class-prior change by distribution matching, *Proceedings of the 29th International Conference on Machine Learning* (2012).
6. Saerens, M., Latinne, P., and Decaestecker, C.: Adjusting the outputs of a classifier to new a priori probabilities: A simple procedure. *Neural Computation*, 14(1), 21–41 (2002)
7. Shimodaira, H.: Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90(2), 227–244 (2000)
8. Twitter, `http://twitter.com/`
9. Vapnik, V. N.: *Statistical Learning Theory*, Wiley, New York (1998)

An Analysis of Interactions Within and Between Extreme Right Communities in Social Media

Derek O'Callaghan[1], Derek Greene[1], Maura Conway[2], Joe Carthy[1], Pádraig Cunningham[1]

[1] School of Computer Science & Informatics, University College Dublin,
{derek.ocallaghan,derek.greene,joe.carthy,padraig.cunningham}@ucd.ie
[2] School of Law & Government, Dublin City University,
maura.conway@dcu.ie

Abstract. Many extreme right groups have had an online presence for some time through the use of dedicated websites. This has been accompanied by increased activity in social media websites in recent years, which may enable the dissemination of extreme right content to a wider audience. In this paper, we present exploratory analysis of the activity of a selection of such groups on Twitter, using network representations based on reciprocal follower and mentions interactions. We find that stable communities of related users are present within individual country networks, where these communities are usually associated with variants of extreme right ideology. Furthermore, we also identify the presence of international relationships between certain groups across geopolitical boundaries.

Keywords: network analysis, social media, community detection, Twitter, extreme right

1 Introduction

Groups associated with the extreme right have maintained an online presence for some time [1], where dedicated websites have been employed for the purposes of content dissemination and member recruitment. Recent years have seen increased activity by these groups in social media websites, given the potential to access a far wider audience than was previously possible. In this paper, we present exploratory analysis of the activity of a selection of these groups on Twitter, where the focus is upon groups of a fascist, racist, supremacist, extreme nationalist or neo-Nazi nature, or some combination of these. Twitter's features enable extreme right groups to disseminate hate content with relative ease, while also facilitating the formation of communities of users around variants of extreme right ideology. Message posts (*tweets*) by such groups are often used to redirect users to content hosted on external websites, for example, dedicated websites managed by particular groups, or content sharing websites such as YouTube.

The first objective of this analysis is the detection of communities of users associated with extreme right groups within individual countries, using network

representations of user interactions. For the purpose of this exploratory work, we have retrieved Twitter data for a selection of eight countries. Having ranked communities detected within each country network representation based on their stability, we generate corresponding descriptions that may be used to provide an interpretation of the underlying community ideology. Our second objective is the identification of international relationships between certain groups that transcend geopolitical boundaries, using two network representations of the interactions between the core users from the eight country sets. It appears that a certain number of international relationships exist, where linguistic and geographical proximity are highly influential.

In Section 2, we provide a description of related work based on the online activities of extremist groups. The generation of the Twitter data sets is then discussed in Section 3. Next, in Section 4, we describe the detection of local extreme right communities within individual countries using two case studies based on the USA and Germany networks. Analysis of the international relationships between extreme right groups from the eight countries is presented in Section 5. Finally, conclusions and suggestions for future work are discussed in Section 6.

2 Related Work

The online activities of different varieties of extremist groups including those associated with the extreme right have been the subject of a number of studies. For example, Burris et al. [1] used social network analysis to study a network based on the links between a selection of white supremacist websites. Chau and Xu [2] studied networks built from users contributing to hate group and racist blogs. The potential for online radicalization through exposure to jihadi video content on YouTube was investigated by Conway and McInerney [3].

Tateo [4] analyzed groups associated with the Italian extreme right using networks based on links between group websites. Caiani and Wagemann [5] studied similar Italian groups along with those from the German extreme right. In their review of the conservative movement in the USA, Blee and Creasap [6] discuss the engagement in online activity as part of an overall mobilization strategy by the more extremist groups within it. Simi [7] analyzed the apparent decentralization of white supremacist groups according to, among other aspects, their online activity. As the majority of this related work involved the study of dedicated websites managed by extreme right groups, we felt that an analysis of their activity in social media would complement this by providing additional insight into the overall online presence of these groups.

3 Data Retrieval with User Curation

Twitter data was collected to facilitate the analysis of contemporary activity by extreme right groups. As the hypothesis was that extreme right communities within social media would tend to be relatively smaller than mainstream communities, the data was sampled using *core sets* of relevant user accounts, one set

per country of interest. A description of this process is available in the extended version of this paper [8].

4 Local Community Detection

The interactions between users within the individual country sets were analyzed with the objective of detecting communities of related users. At the country level, an interaction is defined as one user "mentioning" another with the inclusion of "*@<username>*" within a tweet, where reciprocal mentions between users can be considered as a dialogue [9], thus potentially indicating the presence of a stronger relationship. A network is created with n nodes representing users, and undirected weighted edges representing reciprocal mentions between pairs of users, with weights based on the number of mentions. Currently, all mentions occurrences are considered rather than selecting those from a specific time period. Any connected components of size < 10 are filtered at this point.

We use the method of Greene et al. [10], which is a variant of the recent work by Lancichinetti & Fortunato [11], to generate a set of stable *consensus communities* from a reciprocal mentions network, where 100 runs of the OSLOM algorithm [12] are used to generate the consensus communities. In this analysis, we are primarily concerned with the detection of the most "significant" communities with the strongest signals in the network. Therefore, the consensus communities are ranked based on the stability of their members with respect to the corresponding consensus matrix. We employ the widely-used adjustment technique introduced by [13] to correct for chance agreement:

$$\text{CorrectedStability}(C) = \frac{\text{Stability}(C) - \text{ExpectedStability}(C)}{1 - \text{ExpectedStability}(C)} \tag{1}$$

A value close to 1 will indicate that C is a highly-stable community. As higher values of the threshold parameter τ used with the consensus method result in sparser consensus networks, and having tested with values of τ in the range $[0.1, 0.8]$, we selected $\tau = 0.5$ as a compromise between node retention and more stable communities. Finally, descriptions are generated using a TF-IDF vector for each community, where the terms are hashtags contained within tweets posted by users within the community. A description consists of the top ten hashtag terms ranked using their TF-IDF values. Details of the consensus communities found in the USA and Germany case study networks can be found in Table 1.

Table 1. Consensus communities in local reciprocal mentions networks.

Country	*Original network*		*Consensus network*		*Communities*	*Core users*
	Nodes	*Edges*	*Nodes*	*Edges*		
USA	835	2501	672	6876	55	29
Germany	247	646	167	799	18	46

4.1 Case Study: USA

Table 2. USA reciprocal mentions - selected consensus communities ($\tau = 0.5$)

Communities with core users				
Id	Description	Size	Core (%)	Score
A	aryan, pipa, thewhiterace, shtf, wpww, masterrace, londonriots, ukriots, wwii, nazi	11	2 (18%)	0.89
B	wpww, tcot, gop, truth, teaparty, trayvontruth, blackpower, sopa, mlk, treyvon	24	2 (8%)	0.81
C	ubnp, wpradio, contest, wpww, pandora, wpwwgiveaway, nevershouldyouever, hch, rahowa, survival	75	6 (8%)	0.81
D	prayforthis, glory, genocide, diversity, africans, antiracist, equality, asia, asians, mustsee	11	3 (27%)	0.81
Communities without core users				
Id	Description	Size	Core (%)	Score
E	zumaspear, da, sa, afrikaans, anc, ancyl, zuma, southafrica, malema, afrikaners	38	0 (0%)	0.91
F	rochdale, brighton, edl, mfe, uaf, islam, luton, dewsbury, bbcsml, labour	33	0 (0%)	0.85

A selection of relevant communities having high stability scores can be found in Table 2. Community A would appear to be national socialist/white power in nature, with the appearance of hashtags such as "aryan", "thewhiterace", "wpww" (white pride world wide), "masterrace" and "nazi". An analysis of the users and associated profiles finds references to the *American Nazi Party*, along with other related terms such as *14* (a reference to a 14-word slogan coined by the white supremacist David Lane), and *88* ("H" is the 8th letter in the alphabet, and *88* signifies "heil hitler") in user names. Users appear to be mostly from the USA, although a small number of European users are also present ("londonriots", "ukriots"). Some similar white power themes (e.g. "rahowa" - RAcial HOly WAr) appear in communities B and C, where both communities contain a number of North American users including some who are promoting external white power radio station websites.

Most of the users in community D (both North American and European) appear to be connected with a number of *white rabbit* websites which allege the existence of "white genocide". Communities E and F are interesting as neither community contains a single user from the USA core set. However, an analysis of the users in both communities shows that reciprocal follower relationships with USA users are common. The former appears to consist of white South African users where some profiles contain racist and national socialist references, while the latter contains many references to the *English Defence League* (EDL), a group opposed to the alleged spread of militant islamism within the UK.

4.2 Case Study: Germany

A selection of relevant communities having high stability scores can be found in Table 3. An analysis of the users in community A finds them to be primarily associated with the town of Geithain, near Leipzig in Sachsen ("geithainer", "geithain", "gha" and "lvz" hashtags). Users belonging to various extreme right groupings are present, such as *Freies Netz* (FN - "information portal" websites hosting related content) and the *Junge Nationaldemokraten* (JN -

Table 3. Germany reciprocal mentions - selected consensus communities ($\tau = 0.5$)

Communities with core users				
Id	Description	Size	Core (%)	Score
A	bollywood, lvz, geithainer, geithain, tdi, unsterblichen, gha, mephisto, imc, jcsyhra	10	3 (30%)	0.89
B	saalfeld, gera, pcrecords, apw, 13februar, volkstod, otz, geraer, altenburger, rfd	14	5 (36%)	0.79
C	130abschaffen, unibrennt, raz09, mobilisierungsvideo, golf, demokraten, spreelichter, meilederdemokratie, sfb, halloween	10	2 (20%)	0.77
D	guben, spreelichter, apw, jingle, weltanschauung, vetschau, bock, 17august, podcast, altermedia	24	10 (42%)	0.72
Communities without core users				
Id	Description	Size	Core (%)	Score
E	bamberg, flugblatt, ovg, trke, stolberg, ruhrgebiet, wattenscheid, vorstellungs-flugblatt, rat, landesgartenschau	8	0 (0%)	0.94

Young National Democrats, youth wing of the extreme right *Nationaldemokratische Partei Deutschlands - National Democratic Party of Germany - NPD*). The "unsterblichen" (immortals) hashtag refers to anti-democratic flashmob marches that were occurring sporadically throughout Germany. These protests were linked to *Spreelichter*, an extreme right group that was recently banned by the local authorities. who used social media to propagate national socialist-related material.

Community B also appears to be mostly related to a geographical location, namely the federal state of Thüringen ("altenburg", "gera", "saalfeld"). The "rfd" hashtag refers to "Rock für Deutschland"; a concert organised by the NPD in Gera. Other relevant hashtags include "apw" (außerparlamentarischer Widerstand - non-parliamentary resistance), "13februar" (1945 bombing of Dresden by Allied forces) and "volkstod" (perceived destruction of German people and traditions). The theme of community C is related to the "130abschaffen" hashtag, which refers to demands for the abolition of a paragraph in the German penal code associated with the criminalization of incitement to hatred, along with denial and/or justification of the Holocaust and national socialist rule.

Community D contains a wide range of users from groups such as FN, JN and *Spreelichter/Unsterblichen*. These users would appear to be quite active, with many tweets containing URLs linking to content hosted on external websites such as YouTube or other dedicated websites. No core users are present in community E, but all members appear to be related to the NPD. The hashtags "bamberg" and "ruhrgebiet" refer to locations within Germany, and the presence of other hashtags such as "flugblatt" (flyer/leaflet/pamphlet) and "rat" (council/councillor) along with separate analysis of the tweet content may indicate mobilization prior to elections.

Some further details of the communities from both case studies can be found in the extended version of this paper [8].

5 International Relationships

We also analyzed the international relationships between the various groups within the data sets, based on the interactions between the core users from the

eight country sets. Two types of undirected network were generated; a followers network consisting of user nodes and unweighted edges representing follower links between users, and a mentions network similar to those described in the previous section. As with the country-based networks, only reciprocal edges were used in order to capture stronger relationships, all stored follower and mentions instances were included, and connected components of size < 10 were filtered.

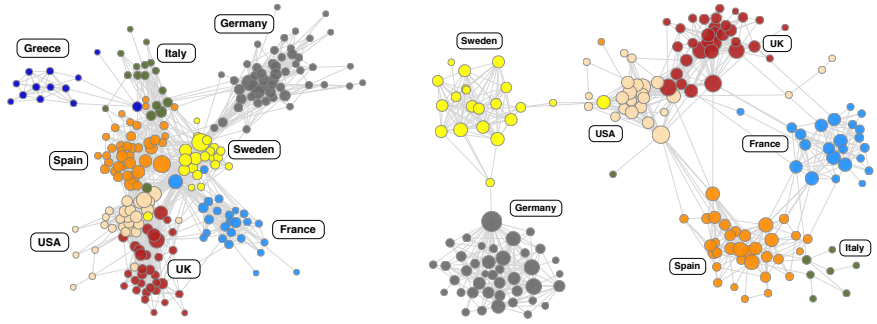


Fig. 1. International core networks: followers (left - 226 nodes, 1603 edges) and mentions (right - 184 nodes, 856 edges). Node size is proportional to degree.

5.1 International Follower Awareness

The international followers network can be seen in Fig. 1 (left). As might be expected, most of the follower relationships are between users from the same country, although a certain number of international relationships are identifiable. It would appear that linguistic and geographical proximity is influential here; similar behaviour with respect to social ties in Twitter was identified by Takhteyev et al. [14]. However, there appear to be some exceptions to the influence of geographical proximity, most notably, Swedish (yellow) and Italian (green) users that are not co-located with their respective country nodes. In both cases, the majority of tweets from these users are in English, which presumably ensures a wider audience.

From an analysis of other central nodes in the network (using betweenness centrality), it would seem that those involved in the dissemination of material (text, audio, video) with the use of external URLs, or media representatives such as extreme right news websites and radio stations, are attempting to raise awareness amongst a variety of international followers. This is especially the case when the English language is used.

5.2 International Dialogue

It is reasonable to consider that the follower-based networks represent passive relationships when compared with the mentions-based networks, where links can

represent actual dialogue between users. The mentions network in Fig. 1 (right) can be seen to be somewhat smaller than the corresponding followers network in Fig. 1 (left); for example, none of the Greek core users are present. Apart from this, the network has a similar structure to that of the followers network, in that most interaction occurs within individual country-based communities. Connections between these communities do exist, but are fewer than in the followers network. The influence of linguistic proximity appears to take precedence here, with the use of English playing a major role as mentioned in the previous section. In the case of the German community, while the followers network contains a number of connections with other international users, this has now been reduced to a single connection with a user acting as an English-language Twitter channel for a Swedish nationalist group. Similarly, the Swedish user co-located with the USA community is the same user as that in the followers network, who appears to be involved in many English-based dialogues.

It should also be emphasized at this point that this analysis does not necessarily provide extensive coverage of the international relationships between all extreme right groups that are active on Twitter. As it is possible that the data retrieved from Twitter is incomplete, the objective is to demonstrate the existence of these relationships by means of an exploratory analysis. This caveat also applies to the country-based community analysis.

6 Conclusions and Future Work

Extreme right groups have become increasingly active in social media websites such as Twitter in recent years. We have presented an exploratory analysis of the activity of a selection of such groups using network representations based on reciprocal follower and mentions interactions. The existence of stable communities of associated users within individual countries has been demonstrated, and we have also identified international relationships between certain groups across geopolitical boundaries. Although a certain awareness exists between users based on the follower relationship, it would appear that mentions interactions indicate stronger relationships where linguistic and geographical proximity are highly influential, in particular, the use of the English language. In relation to this, media user accounts such as those associated with extreme right news websites and radio stations, along with external websites hosting content such as music or video, are a popular mechanism for the dissemination of material among users from a variety of disparate groups.

Although some of the detected communities can be associated with a specific extreme right group or ideology, this is less clear in other cases where communities appear to contain members from a variety of known groups. This may be a consequence of incompleteness in the data sets retrieved for this analysis. It may also be related to variances in Twitter usage patterns in different countries. The laws of different countries should also be taken into consideration, as an opinion that may be legally voiced in one country may not be permitted in another, particularly within the context of extreme right ideals. However, it may also be the

case that social media websites are merely used by such groups to disseminate related material to a wider audience, with the majority of subsequent interaction occurring elsewhere.

In future work, we will address the issue of incompleteness in the data sets, including the current disparity in core set sizes. Local community analysis of countries other than the USA and Germany will be performed. We also plan to study the temporal properties of these networks which will provide insight into the evolution of extreme right communities over time.

Acknowledgements. This research was supported by 2CENTRE, the EU funded Cybercrime Centres of Excellence Network and Science Foundation Ireland Grant 08/SRC/I1407 (Clique: Graph and Network Analysis Cluster).

References

1. Burris, V., Smith, E., Strahm, A.: White Supremacist Networks on the Internet. Sociological Focus **33**(2) (2000) 215–235
2. Chau, M., Xu, J.: Mining communities and their relationships in blogs: A study of online hate groups. Int. J. Hum.-Comput. Stud. **65**(1) (January 2007) 57–70
3. Conway, M., McInerney, L.: Jihadi Video and Auto-radicalisation: Evidence from an Exploratory YouTube Study. In: Proceedings of the 1st European Conference on Intelligence and Security Informatics. EuroISI '08, Berlin, Heidelberg, Springer-Verlag (2008) 108–118
4. Tateo, L.: The Italian Extreme Right On-line Network: An Exploratory Study Using an Integrated Social Network Analysis and Content Analysis Approach. Journal of Computer-Mediated Communication **10**(2) (2005) 00–00
5. Caiani, M., Wagemann, C.: Online Networks of the Italian and German Extreme Right. Information, Communication & Society **12**(1) (2009) 66–109
6. Blee, K.M., Creasap, K.A.: Conservative and Right-Wing Movements. Annual Review of Sociology **36** (2010) 269–286
7. Simi, P.: Why Study White Supremacist Terror? A Research Note. Deviant Behaviour **31**(3) (2010) 251–273
8. O'Callaghan, D., Greene, D., Conway, M., Carthy, J., Cunningham, P.: An Analysis of Interactions Within and Between Extreme Right Communities in Social Media (extended version). ArXiv e-prints (http://arxiv.org/abs/1206.7050) (June 2012)
9. Macskassy, S.: On the Study of Social Interactions in Twitter. In: Sixth International AAAI Conference on Weblogs and Social Media. ICWSM (2012)
10. Greene, D., O'Callaghan, D., Cunningham, P.: Identifying Topical Twitter Communities via User List Aggregation. ArXiv e-prints (http://arxiv.org/abs/1207.0017) (June 2012)
11. Lancichinetti, A., Fortunato, S.: Consensus clustering in complex networks. Sci. Rep. **2** (03 2012)
12. Lancichinetti, A., Radicchi, F., Ramasco, J., Fortunato, S., Ben-Jacob, E.: Finding statistically significant communities in networks. PLoS ONE **6**(4) (2011) e18961
13. Hubert, L., Arabie, P.: Comparing partitions. Journal of Classification (1985) 193–218
14. Takhteyev, Y., Gruzd, A., Wellman, B.: Geography of Twitter networks. Social Networks **34**(1) (2012) 73 – 81

Geographically-Aware Mining of AIS Messages to Track Ship Itineraries

Annalisa Appice, Donato Malerba, and Antonietta Lanza

Dipartimento di Informatica, Università degli Studi di Bari Aldo Moro
via Orabona, 4 - 70126 Bari - Italy
{annalisa.appice, donato.malerba, antonietta.lanza}@uniba.it

Abstract. Spatio-temporal data of a ship moving in an open sea space are mined in order to track the navigation itinerary plausibly followed by the ship until now. A graph analysis of the problem is presented and a geographically-aware itinerary reconstruction strategy is defined. Experiments prove accuracy and efficiency of the proposed solution.

1 Introduction

Marine traffic (http://www.marinetraffic.com/AIS/) is a management system used by the port authorities to know the location of a ship on the Earth's map. The localization is obtained by integrating a Geographical Information System (GIS) with the AIS (automated ship identification) equipment of a ship in order to continuously geo-localize a ship, visualize its trajectory, deduce its behavior also when no transmission is available. By considering that a ship may not necessarily move in straight line between consecutive AIS transmissions (to avoid to cross the mainland), the major challenge of a ship localization system is that of accurately approximating the unknown trajectory (itinerary) between consecutive AIS transmissions. To track continuously the itinerary of a ship, we propose a novel spatio-temporal data mining method which resorts to a best-first graph search strategy and accounts for both the geographical relational information which is spread across the Earth map and the positional information which is enclosed in an AIS message (transmission time, ship position, navigation course and velocity). The paper is organized as follows. In the next Section, we illustrate preliminary concepts of this study. In Section 3, we illustrate the algorithm ShipTracker we have designed to compute and visualize the ship itinerary once a new AIS message is acquired. Finally, we evaluate accuracy and efficiency of the proposed solution and draw conclusions.

2 Background and Preliminary Concepts

We first describe the AIS data and the tessellation data model of the Earth surface. Then we illustrate geographically-aware relations which are considered to characterize any movement of a ship. Finally, we present a way to estimate the itinerary length based on the ship velocity; this estimated length is used as a constraint in the itinerary search.

Marine Traffic and AIS data In Marine Traffic, a ship is associated to its MMSI (the unique numeric code that identifies the ship), nameb and category. For each ship, AIS messages are acquired and stored in a spatio-temporal database. These messages are sent by the transceiver installed on shipboard which sends dynamic messages. Each message includes the ship MMSI; the received time; the latitude and longitude of the ship position; the course over ground and the ship speed.

Tessellation Model and Itinerary In a regular tessellation of the Earth surface, latitude and longitude are both scaled with spatial resolution σ. The base cell of this map tessellation is a square $\sigma \times \sigma$. Each cell is uniquely identified by a row index and a column index of the tessellation matrix [1]. A label is associated to each cell to describe the morphology of the cell surface (land or sea). The ship itinerary flows through sea cells according to the definition reported i nthe following.

Definition 1 (Ship Itinerary). *Let us consider the tessellation of the Earth surface with resolution σ. An itinerary is the sequence of navigable cells, that is:*

$$c_1, c_2, c_3, ...c_n \tag{1}$$

where for each $i = 1 \dots n-1$, $c_{i+1} \in neighborhood(c_i)$.

Geographic-aware knowledge (distance and azimuth relations) The geographical distance is the distance between geographical points measured along the surface of the Earth. In general, the distance between points, which are geo-defined by latitude (la) and longitude (lo), is an element in solving the second (inverse) geodetic problem. Following the main stream of research in several GIS, we resort to simple spherical law of cosines to approximate geographical distance between two points (centroids of cells).

$$d(la_1, lo_1, la_2, lo_2) = R \arccos(\sin la_1 \sin la_2 + \cos la_1 \cos la_2 \cos(lo_2 - lo_1)), \tag{2}$$

where R=6371 is the Earth radius.

The azimuth is an angular measurement in a spherical coordinate system that is calculated by perpendicularly projecting the vector from an observer (origin) to a point of interest onto a reference plane and measuring the angle between it and a reference vector on the reference plane. It is usually measured in degrees (^{o}). In a general navigational context, the reference direction line is the true north, measured as a 0 ^{o} azimuth. The azimuth is used to define the navigation course. We can also use the concept of azimuth to measure the angle of straight line connecting two geographical points with respect to the north line.

$$\alpha(la_1, lo_1, la_2, lo_2) = \arctan \frac{\cos la_2 \sin(lo_2 - lo_1)}{\cos la_1 \sin la_2 - \sin la_1 \cos la_2 \cos(lo_2 - lo_1)}. \tag{3}$$

[1] In this study, we use the global land tessellation of the Earth surface which is publicly provided by the USGS (http://edc2.usgs.gov/glcc/glcc.php) at spatial resolution $\sigma = .00833^o$.

Itinerary Length The length of the unknown itinerary between two AIS messages is estimated based on the time and velocity included in the messages. We assume a constant ship velocity v_c between two consecutive AIS messages. This constant velocity is here estimated as the mean between the intensity of velocity collected in the AIS messages. A velocity based estimate of the length of an itinerary is formulated as follows:

$$length(AIS_1, AIS_2) = v_c \times (time(AIS_2) - time(AIS_1)). \quad (4)$$

3 ShipTracker

Let us consider a ship which sends AIS messages to Marine Traffic by the transceiver installed on shipboard. Once a new AIS message arrives, it is geo-located into an Earth cell denoted as D. Let S be the cell where the previous AIS message was geo-located. ShipTracker starts the discovery process to determine and visualize the unknown itinerary from S until D. The discovery process uses a completely connected graph structure to model the adopted tessellation of the Earth surface. In this way, the cells are represented as nodes of the graph and the geographical relations existing between neighbor cells are naturally modeled by graph edges. Then the task of discovering the unknown itinerary between two consecutive AIS messages is reformulated as the task of looking for the sea labeled path connecting the associated nodes in the tessellation graph.

Several paths may connect two nodes in a graph, hence the challenge of the graph search task is to find quickly the path which fits at the best the unknown itinerary. At this aim, the search procedure is constrained with the estimated itinerary length (computed as reported in Section 2) and the land/sea characterization of the map (described in Section 2). Additionally, we define cost functions which allow us to decide the order at which cells are visited during the search. The definition of a cost function is not trivial in the entire process as it is plausibly influenced by several criteria such as the distance from destination and/or the navigation course.

3.1 Itinerary Search Algorithm

The itinerary search is a recursive process which starts by inputting the presently visited cell C (seed); the destination cell D; and the estimate of the length L of the unknown itinerary i from S to D. Initially, $C = S$ and $L = LMAX$ with $LMAX = length(AIS_S, AIS_D)$ ($LMAX$ is computed according to Equation 4). The itinerary search proceeds recursively by visiting the neighbors of the current seed cell C until the destination cell D has been reached and the itinerary i (from S to C) has been populated. The algorithm returns false if it fails in the itinerary search. At an intermediate step of recursion, C represents the currently visited cell and L is the maximum length of the unknown itinerary we intend to build from C to D. L is determined so that $LMAX - L$ corresponds to the length of the itinerary already computed by visiting the map from S to C. If C

coincides with the destination cell, it is added to the itinerary i and the search stops. If $L < 0$, it means that the length of the computed itinerary until C is over sized with respect to the estimated length of the entire itinerary, hence the itinerary determined until C is not realistic and we backtrack to consider alternative itineraries. Otherwise, the navigable neighborhood of C is determined and neighbors are ranked according to a user defined cost function $h()$. A possible schema according to define h() is described in Section 3.2. The navigable neighborhood of C is the set of sea labeled cells, not yet visited, which surround C in the adopted tessellation of the Earth surface. C is added to the itinerary i and the itinerary search algorithm is recursively called having as visiting seed the cell which is the best ranked in the $Neighborhood$ according to $h()$. If the recursive computation of the itinerary fails by starting from the currently selected neighbor, the search process continues by visiting the remaining neighbors ranked for C until the goal is reached. If the entire neighborhood of C fails in generating the itinerary, the visit removes C from i ("backtracks") as soon as it determines that C cannot complete to a valid solution. Further considerations are necessary to explain the way a neighborhood is determined. We follow here the main inspiration in A* [2] and we expect that a ship traverses a cell by following the itinerary of the lowest known length. This principle allows us to keep, for each cell visited in the map, a sorted priority queue of the alternate itinerary segments which have been explored along the way to arrive to the cell itself. Then we can neglect the exploration of those neighbor cells which are now visited by following an itinerary that is longer than an alternate itinerary already encountered until the same cell. In this case, the cell is purged by the currently computed neighborhood ad it is traversed at the lower-length itinerary segment instead.

3.2 Cost function

Several cost functions have been formulated in order to assign a priority to the neighbors of a cell. The main idea is that several cost functions can be combined in a single one to balance their distinct drawbacks and advantages.

Azimuth based cost (H1) The definition of the cost function $H1()$ founds on the idea that it is highly expected that, in absence of obstacles, a ship will tend to move following the direction of straight-line connecting the current position to the destination position. Let C be the currently visited cell, N be a navigable neighbor of C, D be the destination itinerary cell. Then,

$$H1(C,N) = 1 - \frac{|\alpha(C,N) - \alpha(C,D)}{180} \tag{5}$$

where $\alpha()$ is computed according to Equation 3. H1(C,N) has values in the range [0,1]. The higher values are assigned to the cells which are the best candidates for the inclusion in the itinerary.

Navigation Course based cost (H2) The definition of the cost function $H2()$ takes into account the navigation course (azimuth) reported in the AIS messages. If

there is a drift of the navigation course from the source cell to the destination cell, we can assume that the ship, which initially navigated according to source navigation course, drifted at a time towards the destination navigation course. On the other hand, we expect that the navigation initially proceeds by following the course observed at the source cell while, closer to the destination cell, it proceeds by following the course observed at the destination cell. Let C be the currently visited cell, N be a navigable neighbor of C, S the source cell of the itinerary and D the destination cell of the itinerary. We determine the azimuth differences HCS and HCD. HCS is the absolute difference between the navigation course contained in the AIS message at the source S and the azimuth of the vector connecting the current node C to its neighbor N, that is,

$$HCS(C,N) = 1 - \frac{(|\alpha(AIS_S) - \alpha(C,N)|)}{180}. \tag{6}$$

HCD is the absolute difference computed between the navigation course contained in the AIS message at the destination D and the direction (azimuth) of the straight-line connecting the current node C to N, that is,

$$HCD(C,N) = 1 - \frac{(|\alpha(AIS_D) - \alpha(C,N)|)}{180}. \tag{7}$$

The navigation course is an information enclosed into an AIS message and represents the azimuth of the velocity vector at that time. The cost function H2() is computed as the inverse distance weighed sum of both $HCS(C,N)$ and $HCD(C,N)$, that is,

$$H2(C,N) = \frac{HCS(C,N) \times \frac{1}{d(S,N)} + HCD(C,N) \times \frac{1}{d(N,D)}}{\frac{1}{d(S,N)} + \frac{1}{d(N,D)}}. \tag{8}$$

where $d()$ is computed according to Equation 2. H2(C,N) values range in the interval [0,1]. The higher values are assigned to the cells which are the best candidates for the inclusion in the itinerary.

Geographical distance based cost (H3) The definition of the cost function $H3()$ assigns higher priority to the neighbor cells which are geographically closest to the destination.

$$H3(C,N) = \frac{|d(N,D) - d_m(C)|}{d_M(C) - d_m C} \tag{9}$$

where $d_M(C) = \max_{N \in Neighborhood(C)}(d(N,D))$ and $d_m(C) = \min_{N \in Neighborhood(C)}(d(N,D))$.

H3(C,N) values range in the interval [0,1]. The higher values are assigned to the cells which are the best candidates for the inclusion in the itinerary.

Cost function By considering the pool of cost functions defined above, cells can be really ranked into a neighborhood by using a function $h()$ which is user-defined according to additive schema reported in the following:

$$h(C,V) = \frac{w1H1(C,N) + w2H2(C,N) + w3}{w1 + w2 + w3}. \tag{10}$$

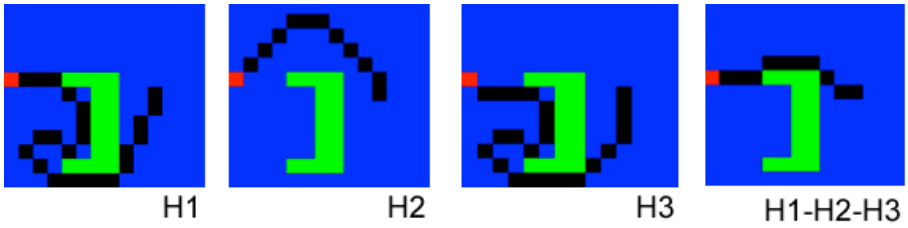


Fig. 1. The itinerary computed by ShipTracker by using h()=H1(), h()=H2(), h()=H3() and h()=(H1()+H2()+H3)/3. The navigation course at the source (red cell) is 45^oN, while the navigation course at the destination is 135^oSW. The number of visited cells ranges from 197 (H1) to 12 (H2), 214 (H3) and 11 (H1+H2+H3).

The choice of weights $w1$, $w2$ and $w3$ is a critical point of the entire process. The example in Figure 3.2 shows that both the itinerary shape and the number of cells visited to compute an itinerary may vary with $h()$. Anyway, this example confirms that, coherently with our formulation of the cost functions, H2() allows us to detect curvilinear itineraries, while the combination of H1, H2 and H3 speeds-up the entire search process.

4 Experiments

ShipTracker is written in Java. We have performed a preliminary set of experiments to evaluate both accuracy and efficiency of itinerary mined from a sample of AIS messages. We run experiments on an Intel(R) Core(TM) 2 DUO CPU $E4500$ @2.20GHz with 2.0 GiB of RAM Memory, running Ubuntu Release 11.10 (oneiric) Kernel Linux $3.0.0-12-generic$.

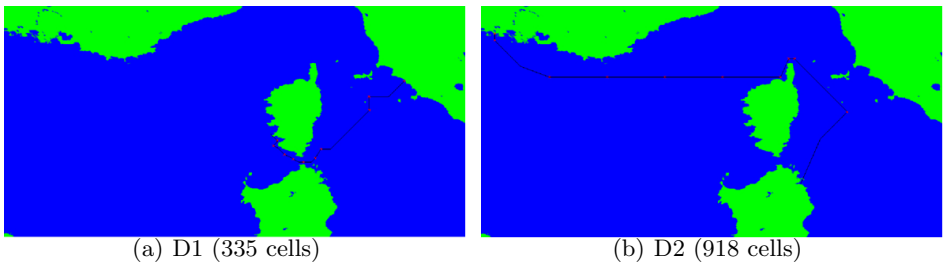
4.1 Data

We consider the two itineraries which are drawn in Figures 2(a)-2(b). They are based on the history of two ships really monitored by Marine Traffic. For each itinerary, we have simulated the AIS message transmitted by the ship each time it crosses a new cell.

4.2 Evaluation Goals and Measures

We study the accuracy and the computation time spent to determine the itinerary of a ship by varying the cost function. The 10% of cells are randomlly chosen in each itinerary to geo-reference AIS messages. To determine the accuracy of an itinerary, we compute the dynamic distance warping (DTW)[2] between [3] the

[2] We base the computation of DTW between the discovered itinerary and the expected itinerary on the implementation provided in the Weka toolkit[1].



(a) D1 (335 cells) (b) D2 (918 cells)

Fig. 2. The ship itineraries of the experimental study.

real itinerary and the discovered. To measure the efficiency of the discovery process, we consider the computation time (in milliseconds) spent to determine the itinerary. Both measures are monitored with a local granularity: the measures under analysis are computed on the sub-itineraries which connect the consecutive AIS messages and then averaged on the entire itinerary. We run ShipTracker by varying the definition of the cost function h according to the combinations H1, H2, H3, H1-H2, H1-H3, H2-H3 and H1-H2-H3.

4.3 Results

The average accuracy and average computation time spent per each sub-itinerary are collected in Tables 1-2. ShipTracker well exploits geographical-aware relations (like azimuth and geographical distance) in order to track an itinerary which is close enough to the expected one. The average computation time spent per sub-itinerary (each itinerary between consecutive AIS massages) is largely below 10 msecs (excepted for H2). This is a competitive computation score we have obtained by exploiting the geographic relational information which is both enclosed in the map (the navigation proceeds by crossing neighboring sea cells which are visitable with a shorter itinerary) and in the AIS messages (the navigation proceeds towards cells with highest priority based on the geographical distance from destination and the azimuth of the navigation course). Final consideration concerns the choice of a cost function. We observe that, in general, the cost H1 in general tracks the more accurate itinerary, while the cost H2 tracks the less accurate itinerary. The cost H3 is in the middle. But if we consider the combination H1+H2+H3, we observe that it is the more accurate after H1, but, in general, it is the most competitive in terms of computation time. We can conclude the cost H2 is useless if not combined with other costs. In particular, the combination represents an acceptable trade-off between accuracy and efficiency.

5 Conclusions

We presented preliminary results of a novel spatio-temporal data mining system which acquires AIS messages transmitted from a ship moving in open sea space,

Table 1. Itinerary D1: average DTW and average computation time (millisecs) by varying the definition of h().

	H1	H2	H3	H1-H2	H1-H3	H2-H3	H1-H2-H3
avgDTW	0.35	1.19	0.62	0.63	0.58	0.64	0.53
avgTime	3.03	36.74	0.97	1.77	0.97	1.00	1.03

Table 2. Itinerary D2: average DTW and average computation time (millisecs) by varying the definition of h().

	H1	H2	H3	H1-H2	H1-H3	H2-H3	H1-H2-H3
avgDTW	0.17	0.42	0.29	0.27	0.24	0.27	0.21
avgTime%	2.22	9.16	1.80	3.47	1.85	1.85	1.80

and use both these data and the map knowledge to visualize the itinerary plausibly followed by the ship between two consecutive AIS messages. The Earth surface is represented by a tessellation of land/sea cells and a graph-search algorithm has been tailore in order to discover the ship itinerary in the form of a graph path which connects the cell nodes of the map tessellation where the AIS messages are geo-located. The geographical-aware relational knowledge which is enclosed both in the map and in the AIS messages (neighboring relations, geographical distance, azimuth) has been considered in the definition of the cost functions incorporated to speed-up the itinerary search in the graph. Preliminary experiments evaluate both accuracy and efficiency of the proposed system and prove the viability of itinerary discovery process also if we drastically reduce the number of the AIS messages.

Acknowledgments

This work fulfills the research objectives of the project PRIN 2009 Project "Learning Techniques in Relational Domains and their Applications" funded by the Italian Ministry of University and Research (MIUR). Authors thank Guido Colangiuli and Rocco Rana for developing a Java implementation of ShipTracker.

References

1. E. Frank, M. Hall, G. Holmes, R. Kirkby, B. Pfahringer, I. H. Witten, and L. Trigg. Weka-a machine learning workbench for data mining. In O. Maimon and L. Rokach, editors, *Data Mining and Knowledge Discovery Handbook*, pages 1269–1277. Springer, 2010.
2. N. J. Nilsson. *Principles of Artificial Intelligence*. Palo Alto, California: Tioga Publishing Company, 1980.
3. H. Sakoe and S. Chiba. Readings in speech recognition. chapter Dynamic programming algorithm optimization for spoken word recognition, pages 159–165. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1990.

Robust Language Identification in Short, Noisy Texts: Improvements to LIGA

John Vogel[1] and David Tresner-Kirsch[1,2]

[1]Brandeis University, Department of Computer Science
415 South Street, Waltham, MA 02453, USA
{locus,dwkirsch}@brandeis.edu
[2]MITRE Corporation
202 Burlington Road, Bedford, MA 01730, USA
davidtk@mitre.org

Abstract. Language identification (LI) is a crucial preprocessing step for natural language processing tasks and other secondary uses of documents from multilingual sources. Conventional machine learning approaches to LI perform very well on long documents using standard language, but relatively poorly on short documents rife with non-standard orthography. This limitation is an obstacle to secondary uses of social media data from Twitter and other similar sources. We propose several linguistically-motivated modifications to the LIGA algorithm, and evaluate these modifications empirically. Our results show that a modified algorithm achieves 99.8% accuracy disambiguating among six European languages, reducing baseline LIGA's error rate by roughly an order of magnitude.

Keywords: Twitter, microblogs, language identification, natural language processing, machine learning

1 Introduction

Language identification (LI) is a text classification step that necessarily precedes many natural language processing methods intended to apply to multilingual data sources. Application of background properties, models, and techniques specialized to a document's language presupposes knowledge of the language in which the document is written.

Cavnar and Trenkle [3] introduce a classification method based on most-frequent n-gram profiles of languages. Language identification of documents is performed by comparing the frequency-ranked lists of n-grams in the language profile to that in the target document. This method is widely used and is 99.8% accurate on documents of longer than 400 characters.

However, there is a growing focus in research and practice on tasks such as information extraction and opinion mining in microdocuments much shorter than 400 characters, due to the exploding quantities of data from social sources including Twitter and SMS. The quantity of mineable social microdocument

data is enormous: in 2010, 175 million authors had Twitter accounts producing in aggregate 65 million tweets per day, and that authorship rate has continued to climb [2]. Messages from both Twitter and SMS are limited to 140 characters, which presents a challenge to text classification tasks. Text processing tasks are further challenged in these domains by a high prevalence of non-standard orthography: short, social messages are rife with abbreviations, ellisions, phonetic substitutions, and even lengthenings [4, 5].

Cavnar and Trenkle's n-gram profile comparison method breaks down for these very short documents. Tromp and Pechenizkiy [6] report that the n-gram profile method acheives only 93.1% accuracy when distinguishing between six languages in Twitter posts (after cleaning the posts by removing #-prefixed keywords, links, and @-prefixed account names). Tromp and Pechenizkiy also introduce LIGA, a graph-based text classification method specifically targetting short texts which acheives an accuracy of 97.5%, a reduction in error rate of more than half.

Bergsma et al. report that a modified version of Cavnar and Trenkle's algorithm acheives 96.3% accuracy disambiguating tweets among nine languages: four Indic languages, three Baltic languages, Arabic, and Farsi [1]. They demonstrate improved accuracy using compression-based language models (97.1%) and maximum entropy classifiers (97.4%) trained solely on Twitter data, and acheive 97.9% accuracy training the compression model on both Twitter and Wikipedia data.

The accuracy scores reported in [6] and [1] are likely sufficient for many tasks. Nonetheless, accuracy in the Twitter domain still lags far behind that for longer documents. The sheer quantity of microblog data makes it desirable and meaningful to further reduce the error rate of state-of-the-art LI algorithms; even an error rate as low as 2.1% means that more than a million newly published tweets would be mislabeled daily by the best existing algorithms.

In this paper, we propose several improvements to the LIGA algorithm, and evaluate these approaches on the same data set used in [6]. Ultimately, we demonstrate improvement upon LIGA's error rate by roughly an order of magnitude.

2 Data

For training and testing, we used a curated Twitter corpus collected for previous work on language identification [6]. The corpus contains 9066 tweets in six European languages (three Germanic and three Romance) from 36 author accounts (distributed evenly among the six languages). In order to create a clean language-labeled gold standard, tweets containing text in multiple languages (by manual determination) were not included in the corpus. The individual tweets have also had Twitter-specific terminology (@-prefixed usernames, #-prefixed keywords, punctuation, and emoticons) removed. After filtering, the average document length is 80 characters (minimum: 12; maximum: 139). The included languages, and the counts of tweets in each language subcorpus, are shown in Figure 1.

Language	German	UK English	Spanish	French	Italian	Dutch	Total
# of documents	1479	1505	1562	1551	1539	1430	9066
Average length	77	85	75	85	78	81	80

Fig. 1. Quantity of data per language

3 Methods

All the work presented here builds off the LIGA algorithm described in [6]. LIGA models texts as graphs in which the nodes are trigrams of characters and the edges are transitions from one character trigram to the next. For example, the text *apple* would be represented in the graph by traversing trigram nodes *app*, *ppl*, and *ple*, and edges (*app*,*ppl*) and (*ppl*, *ple*). See Figure 2.

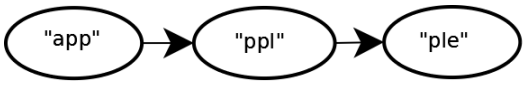


Fig. 2. A basic LIGA graph for the text "apple".

In order to build a language model, LIGA uses a training corpus of texts in that language, and calculates the frequency with which each character trigram and trigram transition (i.e. each node and each edge) appears; the raw count of each node or edge normalized by the total number of nodes or edges in that language. So, for example, if the training corpus of English texts contains the texts *apple* and *an app*, there would be seven total trigrams and five total transitions, which would mean the frequency of the trigram *app* would be $\frac{2}{7}$, the frequency of *ppl* would be $\frac{1}{7}$, and the frequency of the transition (*app*,*ppl*) would be $\frac{1}{5}$.

Once the language models have been built, LIGA classifies text of unknown language by looking at the trigrams and trigram transitions in the text and choosing the language whose model maximizes the total frequency of those trigrams and trigram transitions, thus having the highest correspondence to the unknown text. So, for example, using the training corpus of English texts above, the text *appl* would have a total frequency relative to English of $\frac{2}{7} + \frac{1}{7} + \frac{1}{5}$, or about 0.63. If that was the highest total frequency out of all the language models, the text *appl* would be classified as English.

In this paper we investigate the effects of four different augmentations to this basic LIGA algorithm: fragmenting the graph by word-length, reducing the weight of repeated information, scoring by median, and using log frequencies.

3.1 Word Length Information

The trigram and trigram transition information used by the LIGA model captures the common patterns of letter sequences in each language. However, one reason why a particular pattern of letter sequences might be common in a given language is that that pattern is a common morpheme in that language. For

example, in English *-ness* is a common derivational morpheme that converts adjectives into nouns, so its trigrams *nes* and *ess* and its trigram pair (*nes*,*ess*) will have high frequency in a LIGA model of English. However, this pattern is not otherwise common in English, so seeing trigrams like *nes* and *ess* is only a good indicator that a text is English if the word is actually using the *-ness* morpheme. So rather than modelling "nes" as an indicator of English per se, it may be better to model "nes" as an indicator of English when it is used in a morphologically complex word.

To test this hypothesis, we use word length as a proxy for morphological complexity. We calculate the basic LIGA model as above, then calculate a separate model over the same data. This model uses normalized frequencies just as in the basic LIGA model, except that instead of calculating the frequency of a particular character trigram, we calculate the frequency of a character trigram in a word of a particular length, and instead of calculating the frequency of a particular trigram transition, we calculate the frequency of a trigram transition in a word of a particular length. For trigrams and transitions that span word boundaries, the word length used is the length of the first word. So, for example, in the string *the_apple* (underscore representing whitespace), nodes would be (*the*,3), (*he_*,3), (*e_a*,3), (*_ap*,5), (*app*,5), (*ppl*,5), and (*ple*,5), all with frequency $\frac{1}{7}$. Then, to classify the language of an unknown text, we use both the basic LIGA model and this model augmented with word-length information. The text is classified according to which language has the highest correspondence to the text based on the average of both models. We refer to this model as **lengthLIGA**.

3.2 Reducing the Weight of Repeated Information

The LIGA model assumes that trigrams and trigram transitions within the same text are independent from one another. That is to say, if a particular trigram that indicates the French language shows up twice in a particular text, that's twice as much information in favor of the text being in French than if the trigram had only shown up once. However, that may only indicate that that text is on some particular topic which involves the use of a French loan-word or otherwise abnormally French-seeming word, and that trigram appears in the word. In general, the second appearance of a trigram in a given text may not provide as much information as the first appearance did, because once a word has been used in a text, that makes it much more likely to be used again in that text, as opposed to any other random word. The effect of this is two-fold. First, it means that repeated instances of the same trigram (or transition) in a given training text may provide less information in building a model, since they might be appearing because of the topic of the text rather than the language of the text. Second, it means that repeated instances of the same trigram (or transition) in a given unknown text may provide less information about what language the text is in, for the same reason.

To compensate for the effect of repeated information in a text and test this hypothesis, we build a LIGA model as normal, except that each occurrence of a given trigram or trigram pair in a given training text after the first is treated

as one fifth of an occurrence for calculating frequencies. For example, in the text *aaaah*, trigrams would be *aaa* with frequency $\frac{6}{11}$ and *aah* with frequency $\frac{5}{11}$. In an ordinary LIGA model of that word, *aaa* would have frequency $\frac{2}{3}$ and *aah* would have frequency $\frac{1}{3}$. Similarly, when classifying test texts, repeated occurrences of a particular trigram or trigram pair only added one fifth of their frequency to the score for each language. We refer to this model as **repLIGA**.

3.3 Median Scoring

The LIGA model uses the sum of the frequencies of each trigram and trigram transition in each language's model to determine a given text's language. However, this allows outliers to skew the results when one particular trigram or trigram transition in a text has an abnormally high correspondence to a particular language. Outliers frequently occur when a text in one language uses a loan-word from another language, or refers to a named entity that is named according to the standards of another language (as often happens in reporting of news from other parts of the world). In these cases, the words with foreign language trigrams may overwhelm the other words' correspondence with the text's true language and cause the text to be misclassified.

To protect against outliers skewing the decision and test this hypothesis, we build a LIGA model as normal. However, when determining the language of an unknown text, the classification is made based on maximizing the median trigram frequency plus the median trigram transition frequency over the language models, rather than maximizing the total frequencies. Thus a few trigrams or transitions with especially high or low frequencies in a given language will be ignored, and the decision made solely on the basis of the trigrams and transitions in the middle of the distribution. We call this model **medianLIGA**.

3.4 Logarithmic Frequencies

Zipf's Law [7] states that frequencies of words in a language follow an exponential distribution. This means that the most frequent words in the language will tend to dominate the training data, even if they do not tell the whole story of the patterns of character distribution in the language. This may lead to misclassifications when trigrams and transitions that appear in very frequent words in one language show up in less-frequent words of another language. Texts that have these less frequent words would be misclassified based on their superficial resemblance to common words in another language.

To correct for the exponential distribution of words and test this hypothesis, we build a LIGA model as normal, except that after counting the number of occurrences of each trigram and trigram transition in the training data, we take the natural logarithm of all the counts before calculating the frequencies. This has the effect of converting the exponential distribution into a linear one, and also of eliminating data points that appear only once in the training data, reducing noise. We refer to this model as **logLIGA**.

3.5 Combined Models

Not all of the proposed improvements can be meaningfully combined. For example, using median scoring would render pointless the use of logarithmic frequencies, since mapping frequencies to logarithms would not change their relative order. However, since the **lengthLIGA** model simply changes the shape of the graph, it is easily combinable with models based on scoring alterations. We use two combined models: **medianLengthLIGA** and **logLengthLIGA**.

4 Experiments

To test the effectiveness of each of the proposed improvements to the LIGA system, we use the data from [6]. For each of the six language subcorpora, we randomly choose half of the texts to make up the training data set, leaving the other half as the test set. We then build a model based on the the training data, and evaluate that model's classifications against gold standard labels in the test set. We repeat this process fifty times, choosing a different random split of each language subcorpus for the training and test data each time. We microaveraged the results to get an overall accuracy score for the system, as well as precision and recall information for each language in the data set. The results for baseline LIGA and six experimental systems are presented in Figure 3.

To test the effects of the size of the available training data set on the effectiveness of each of the systems, we vary the fraction of the data used as training data versus testing data. The process of dividing the data set randomly between training and testing data fifty different ways and microaveraging to find accuracy remains the same, the only change is in the relative sizes of the training and testing data sets. These results are presented in Figure 4.

5 Results

In Figure 3 we show the averaged results of experiments with each experimental system. Accuracy is reported for each individual language **L** as precision **P** (# correctly labeled **L** / # labeled **L**), recall **R** (# correctly labeled **L** / # in **L** subcorpus), and **F**-score (harmonic mean of **P** and **R**). We also report the overall accuracy across all languages (# correct / # of documents).

Algorithm	%	French			Italian			Spanish			English			Dutch			German		
	Acc.	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
Baseline LIGA	.979	.986	.954	.969	.984	.984	.984	.990	.970	.979	.939	.997	.967	.990	.988	.988	.985	.984	.984
lengthLIGA	.986	.987	.976	.981	.992	.987	.989	.991	.984	.987	.969	.996	.982	.997	.991	.993	.989	.992	.990
repLIGA	.986	.991	.968	.979	.989	.986	.987	.993	.979	.985	.956	.998	.976	.993	.995	.993	.991	.990	.990
medianLIGA	.988	.990	.993	.991	.995	.978	.986	.984	.996	.989	.987	.979	.982	.986	.995	.990	.990	.993	.991
medianLength	.990	.991	.995	.993	.997	.981	.989	.987	.996	.992	.987	.981	.985	.987	.995	.991	.991	.995	.993
logLIGA	**.998**	.995	.999	.996	>.999	.993	.996	.998	.998	.998	.993	.999	.995	>.999	.999	.999	.999	.998	.998
logLength	**.998**	.995	.999	.997	>.999	.993	.996	.998	.998	.998	.993	.998	.995	>.999	.999	.999	.999	.998	.998

Fig. 3. Accuracy of various algorithms, averaged across 50 random training/test samples

The baseline LIGA system acheived overall accuracy of .979, confirming within the margin of error the accuracy of .975 reported in [6].

All experimental systems outperformed baseline LIGA. They do so consistently across individual languages and metrics, with a few language-specific exceptions in the median-based systems. The highest scoring system was **logLIGA**, which acheived an overall accuracy of .998 (compared to baseline .979). The system **medianLengthLIGA** system, combining length information with median scoring, performed better than individual systems using each of those techniques (**lengthLIGA** and **medianLIGA**, respectively). However, the system combining length information with log-weighting the graph (**logLengthLIGA**) was statistically tied with the individual **logLIGA** system.

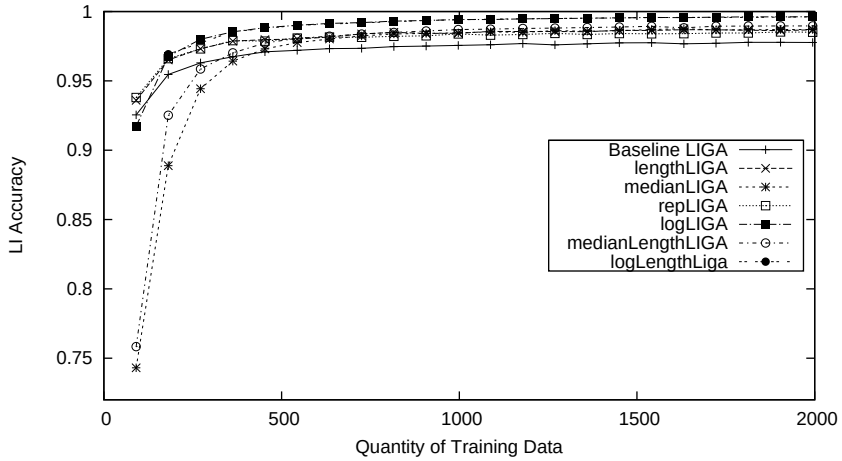


Fig. 4. Results of learning curve experiments

In Figure 4, we show the learning curves resulting from experiments with varied amounts of training data for each system. Although these experiments included runs with up to 7000 training documents, the graph is truncated at 2000 because very little improvement was seen beyond that.

We observe that **medianLIGA** initially learns slower than the baseline and other mean-based systems, even though with continued training it overtakes some of the mean-based systems. This is easily explained: median-based systems will issue a language-match score of zero for any document in which fewer than half of its trigrams and trigram-pairs have been observed in the training data. Unlike mean-based systems, **medianLIGA** will make no distinction between a document in which 40% of trigrams have substantial weights in the graph and one in which only 10% of trigrams appear in the graph. The combined system **medianLengthLIGA** shares this delay in learning for identical reasons.

6 Conclusion

LIGA is already a high-accuracy algorithm; 97.9% accuracy is likely sufficient for many tasks. However, given the scale of data in the Twitter domain, reduction of this 2.1% error rate remains meaningful: more than a million tweets published each day would be misclassified by the baseline algorithm.

We have shown several modified versions of the LIGA algorithm which provide improved performance for language identification in the Twitter domain. In particular, **logLIGA** provides a ten-fold reduction in error rate for negligible computational cost and no increase in memory requirement. Of systems evaluated in this study, it is the clear choice for accurate classification of tweets.

This study was limited by the size and scope of the dataset used. In future work we intend to apply the algorithms presented here to a larger or more diverse set of data, such as the dataset presented in [1], to confirm that the results generalize across a broad range of languages. It will also be valuable to apply the algorithms to other domains of short text, such as SMS texts, to confirm that the results are not specific to Twitter.

Acknowledgments. DTK was funded under the MITRE Innovation Program.

References

1. Bergsma, S., McNamee, P., Bagdouri, M., Fink, C., Wilson, T.: Language identification for creating language-specific twitter collections. In: Proceedings of the Second Workshop on Language in Social Media. pp. 65–74 (2012)
2. Burger, J.D., Henderson, J., Kim, G., Zarrella, G.: Discriminating gender on twitter. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. pp. 1301–1309. EMNLP, ACL (2011), http://dl.acm.org/citation.cfm?id=2145432.2145568
3. Cavnar, W.B., Trenkle, J.M.: N-gram-based text categorization. In: Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval. pp. 161–175 (1994)
4. Han, B., Baldwin, T.: Lexical normalisation of short text messages: Makn sens a #twitter. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. pp. 368–378. Association for Computational Linguistics, Portland, Oregon, USA (June 2011), http://www.aclweb.org/anthology/P11-1038
5. Sproat, R., Black, A.W., Chen, S., Kumar, S., Ostendorf, M., Richards, C.: Normalization of non-standard words. Computer Speech and Language 15(3), 287 – 333 (2001), http://www.sciencedirect.com/science/article/pii/S088523080190169X
6. Tromp, E., Pechenizkiy, M.: Graph-based n-gram language identification on short texts. In: Proceedings of the Twentieth Belgian Dutch Conference on Machine Learning (Benelearn). pp. 27–34 (2011)
7. Zipf, G.K.: Selective Studies and the Principle of Relative Frequency in Language. Harvard University Press (1932)

Identifying Time-Respecting Subgraphs in Temporal Networks*

Ursula Redmond, Martin Harrigan, and Pádraig Cunningham

School of Computer Science & Informatics
University College Dublin
ursula.redmond@ucdconnect.ie,{martin.harrigan,padraig.cunningham}@ucd.ie

Abstract. In the analysis of temporal networks, much information is lost if they are observed as static or as a sequence of static snapshots within time windows. Examining the interactions between individuals with all of the temporal information intact gives a more meaningful understanding of the processes represented by a network. Sections of a temporal network which facilitate flow – be it of information, money or infection – we term time-respecting subgraphs. We propose an algorithm to identify time-respecting subgraphs, and present the results of its application to a temporal network.

1 Introduction

A vast collection of systems can be modeled as graphs, with actors represented by vertices and their interactions as edges. Many of these graph structures provide the infrastructure for some dynamic system, in which the edges are not continuously in use, but rather facilitate interactions among actors for specified periods of time. This motivates the inclusion of temporal structure, as well as topological structure, in the study of dynamic systems which support processes such as disease contagion and information diffusion. In a temporal graph, the time when edges are active is an explicit element of the representation [1].

Given a temporal graph, we can ask such questions as: Who is the most central actor in a network in which individuals are linked only for the duration of a communication? Can we follow a piece of information as it is transmitted through a communication network? In an air-transport network, which are possible sequences of connecting flights [2]? In a financial network, do specific transaction patterns among groups of individuals correspond to fraudulent activity?

To answer these questions and others satisfactorily, the analysis of subgraphs representing the activity of the individuals of interest is revelatory. Since the connections between actors often take many forms, examining only the most important path between them is very limiting. It may be difficult to define which path is the most important, if one exists at all. Thus, analyzing a subgraph gives a greater chance that the most import path is present [3].

* This work is supported by Science Foundation Ireland under Grant No. 08/SRC/I1407.

If the network under study is composed of relatively constant relations, the static approach to identifying reachable vertices is adequate. However, if relations change quickly, relative to the transmission time of the interaction, or network contact occurs at discrete moments, then network measures of reachability must account for this dynamism [4]. Assuming that a temporal network is completely concurrent implies the greatest possible diffusion potential. The actual diffusion paths are a subset of all of these possible paths, so any measure calculated on the static interpretation of this graph will be incorrect and potentially misleading.

Thus, we introduce a formulation for identifying time-respecting subgraphs. Within these, the flow between vertices occurs via sequential interactions, with the duration of and the time between interactions playing an explicit part in the identification process. We apply our proposed algorithm to network data derived from the Prosper Marketplace [5], an online peer-to-peer lending system.

The related work motivating the contributions of this paper is provided in Section 2. A definition of time-respecting subgraphs that captures sequences of interactions with temporal restrictions, along with an algorithm to identify these, is presented in Section 3. Section 4 discusses the results of our application of the algorithm to a temporal network of financial transactions. The paper concludes with a summary of our contributions and suggestions for future work.

2 Related Work

A common approach to account for the temporal dimension of a network is to segment the data into time windows, often with multiple interactions aggregated into single edges. The evolution of the network is then evaluated with respect to these windows, within which each network is taken as static. However, in a static network, given a directed edge between vertices u and v, and between v and w, then u is indirectly connected to w via a path through v. The same cannot be said for a temporal network. In this case, if the directed edge (v, w) is active before the edge (u, v), then u and w are disconnected, since nothing can propagate between them via v.

Another problem arising in the case of aggregated temporal networks is the choice of time window size. Certain aggregations may fail to capture network properties of interest. For example, in a network of mobile telephone calls, networks aggregated over different time intervals yielded different insights [6]. It was seen that for short time windows, the cluster and community structures dominate, whereas for longer windows, the contribution of weak and random links increases.

Many concepts which are well understood for static graphs, but must be revised with the introduction of temporal information, are introduced in a comprehensive review by Holme *et al.* [1]. The ideas most relevant for this paper are here introduced.

In a temporal graph, Kempe *et al.* [7] define a *time-respecting* path as a sequence of contacts with non-decreasing times that connect sets of vertices. According to Nicosia *et al.* [8], two vertices i and j are *strongly connected* if there

is a directed, time-respecting path connecting i to j and vice versa. Vertices i and j are *weakly connected* if there are undirected time-respecting paths from i to j and vice versa. In this case, the directions of the contacts are not observed [8].

In a reachability graph, a directed edge exists between vertices i and j if there is a time-respecting path between them. The algorithm supplied by [4] to identify reachability graphs treats all vertices as seeds, from which reachable graphs are grown in a breath-first-search manner. This reveals the vertices which are reachable from each other. Another name for this type of graph is the *associated influence digraph* of a time-stamped graph, which encodes the ability of vertices to influence, or reach, each other [9].

Bearman *et al.* [10] analyze a reachability graph constructed from a dating network of high-school students. In this temporal framework, the authors demonstrate that such networks are characterized by long chains of contact and few cycles, contrary to what was expected.

From the field of epidemiology, the notion of a transmission graph emerged [11]. This expands the concept of a line graph to capture temporal information. A parameter to measure the incubation time and duration of a disease is incorporated in the construction of a transmission graph. The representation is derived from an interval graph, in which an edge is active over a specified interval. An edge in the transmission graph exists if two edges, which follow each other in time, share a vertex through which the disease may be spread.

In the formulation of Zhao *et al.* [12], the lifespan of a piece of information is specified by a time window. This window measures the time between the end of one communication and the beginning of another. The authors define a communication graph from a subset of records in a call detail record data set, in which each call is connected to at least one other within the time window. The definition provided is in the same vein as what we propose in this paper, although their version is slightly looser, admitting combinations of interactions that we do not regard as time-respecting.

In another work, the *relay time* of an edge is defined as the time taken for a newly infected node to further spread the infection via the next interaction that the link participates in [13]. Thus, limiting the time allowed between interactions is an important concept in a variety of network types.

Correspondingly in this work, we require the time delay between contacts on a time-respecting path not to exceed a threshold d. To see why, consider a shortest path between vertices i and j via a vertex k. The centrality of the vertex k is vulnerable, in that it depends on the interval between the communication from i to k and from k to j. The longer this interval, the higher the chance that the information intended for j will be disrupted [14]. Also, although it is possible that there is no causality relationship between any two adjacent communications, it appears that the closer in time they take place, the more likely they are to be about the same topic [12].

3 Identifying Time-respecting Subgraphs

Since there have been varied approaches to the representation of the temporal dimension in network studies thus far, as noted in Section 2, a precise definition of our view of a directed temporal network follows.

Definition 1. *A directed temporal graph G consists of a set V of vertices and a set E of ordered pairs of vertices representing interactions. An interaction e_i in E is represented by a four-tuple $e_i = (u_i, v_i, t_i, \delta_i)$, in which u_i is the source vertex, v_i is the target vertex, t_i is the initiation time of the interaction and δ_i is the duration of the interaction.*

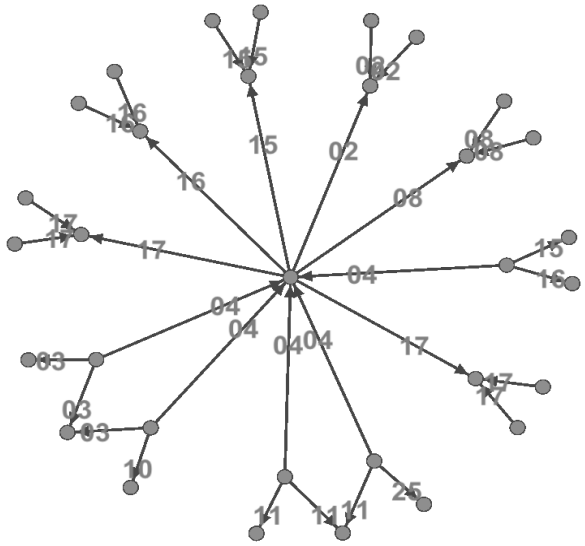


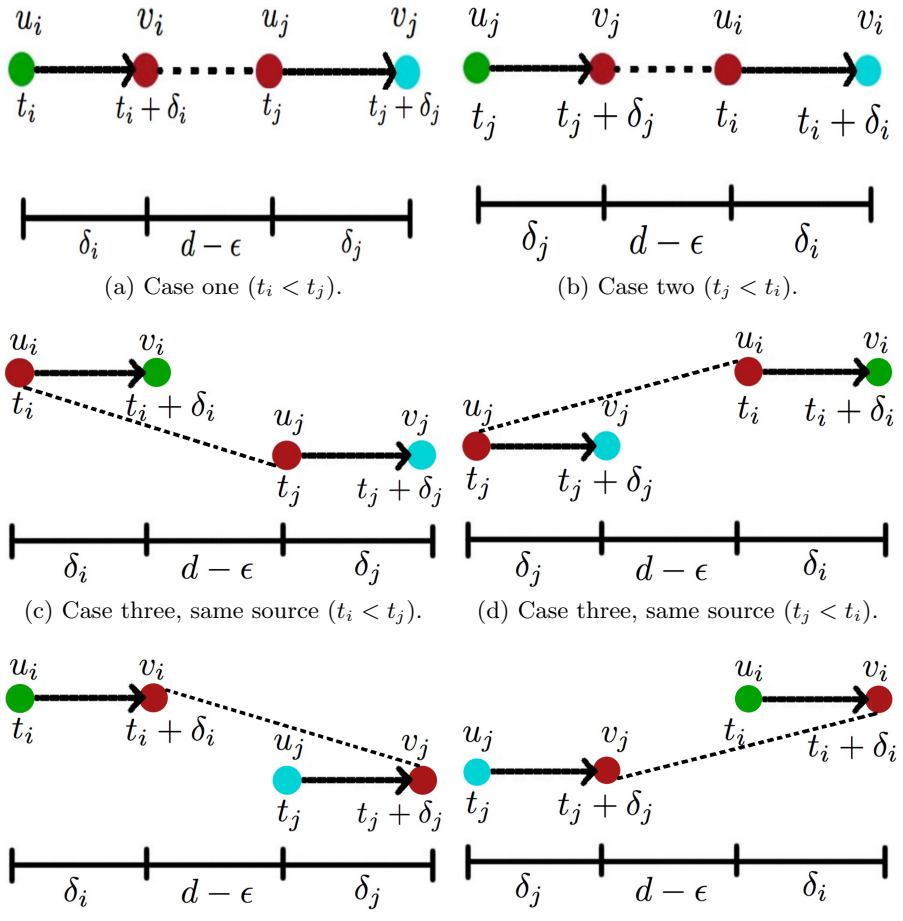
Fig. 1: Time-respecting subgraph, with the day of occurrence stamped on the interactions.

Here we present the notion of a time-respecting subgraph. A motivating example from a larger subgraph, extracted from a real network, is shown in Figure 1. The image – and other examples to follow – represents a sample from the ego-centric network of an individual with incoming and outgoing interactions in a time-respecting subgraph, since the entire subgraph was too large to visualize. Indeed, individuals acting thus as both lenders and borrowers have very large and dense ego-centric networks, so only a sample from their networks could be shown. The possible pair-wise constructions of such a subgraph are illustrated in Figure 2.

Definition 2. *A time-respecting subgraph $R = (V', E')$ of a temporal graph $G = (V, E)$ is composed of a vertex set $V' \subseteq V$, and a set of interactions $E' \subseteq E$ such that $\forall e_i \in E'$, there exists at least one interaction $e_j \in E'$ with $e_i \neq e_j$ such that*

(i) $0 < |\{u_i, v_i\} \cap \{u_j, v_j\}|$

(ii)
$$\begin{cases} 0 \leq t_j - t_i - \delta_i \leq d & \text{if } v_i = u_j \\ 0 \leq t_i - t_j - \delta_j \leq d & \text{if } u_i = v_j \\ 0 \leq t_j - t_i - \delta_i \leq d \text{ or } 0 \leq t_i - t_j - \delta_j \leq d & \text{otherwise.} \end{cases}$$



(a) Case one ($t_i < t_j$). (b) Case two ($t_j < t_i$).

(c) Case three, same source ($t_i < t_j$). (d) Case three, same source ($t_j < t_i$).

(e) Case three, same target ($t_i < t_j$). (f) Case three, same target ($t_j < t_i$).

Fig. 2: Examples to illustrate the application of conditions (i) and (ii).

Definition 2 requires that there be a shared vertex between a pair of interactions in the same time-respecting subgraph. The three cases specified by condition (ii) arise with respect to how the interactions are connected, as shown in Figure 2.

Figure 2a illustrates the first case. Vertices v_i and u_j are the same, with different labelings within their individual scopes. Thus, condition (i) is satisfied. The start and end times of each interaction are labeled, and the time scale is also shown on the line underneath. It can be seen that $t_j - (t_i + \delta_i) = d - \epsilon$, where $0 < \epsilon$. Simplifying this expression, we have that $t_j - t_i - \delta_i = d - \epsilon \leq d$ and condition (ii) is satisfied. An analogous argument can be made regarding the second case, as shown in Figure 2b, in which vertices v_j and u_i are the same.

The remainder of the examples cover the third case, in which a pair of interactions share either a source vertex or a target vertex. In these cases, there is no restriction on the order of the interactions initiated by the source vertex or finishing at the target vertex. To take the example in Figure 2f, the target vertices v_j and v_i are the same, which satisfies condition (i). The interaction e_j occurrs before e_i. Thus, we have that $t_i - (t_j + \delta_j) = d - \epsilon$, where $0 < \epsilon$. This yields $t_i - t_j - \delta_j = d - \epsilon \leq d$, satisfying condition (ii).

3.1 Algorithm

The classical application of a breadth-first search (BFS) traverses the vertices of a graph. It begins with a root vertex, inspects all neighbouring vertices, then inspects their neighbours in turn, and so on until some finishing condition is met. The connected components of a graph may be computed in linear time, using repeated applications of BFS from vertices not already included in a previously found connected component [15].

In the search for time-respecting subgraphs, we employ a similar methodology. A BFS traverses the interactions of a temporal network, rather than the vertices. It begins with a root interaction, and inspects all adjacent interactions (these fulfill condition (i) of the definition). Those that comply with condition (ii), with respect to the root interaction, are included in the subgraph. Accepted interactions undergo the same expansion procedure, and so on until the subgraph can no longer be expanded. All maximal time-respecting subgraphs are found using repeated applications of the BFS algorithm, as outlined in Algorithm 1, starting from interactions not already part of a previously found subgraph.

In accordance with the definition of a time-respecting subgraph, there are three cases in which a pair of interactions may be deemed time-respecting. A separate function is called to evaluate each of these circumstances for each interaction. The first two cases deal with the source vertex of one interaction being the same as the target vertex of another interaction. The final case deals with a pair of interactions sharing either a source or target vertex. Interactions complying with conditions (i) and (ii) are included in the next iteration of the breadth-first-search.

Algorithm 1 Find Maximal Time-respecting Subgraph (G, e, d)

function bfs_augmented(G, e, d)
$q \leftarrow e, s \leftarrow e$
while q is non-empty **do**
  $t \leftarrow q.dequeue()$
  adj_edges $\leftarrow$ out_target(G, t, d) + in_source(G, t, d) + otherwise(G, t, d)
  for all $o \in adj\_edges$, $o \notin s$ **do**
    $q \leftarrow o, s \leftarrow o$
  end for
end while
return s
end function

function out_target(G, e_i, d)
$out\_target \leftarrow \emptyset, v_i \leftarrow e_i$.target()
for all $e_j \in G.out\_edges(v_i)$ **do**
  if $0 \leq t_j - t_i - \delta_i \leq d$ **then**
    out_target $\leftarrow e_j$
  end if
end for
return out_target
end function

function in_source(G, e_i, d)
in_source $\leftarrow \emptyset$, $u_i \leftarrow e_i$.source()
for all $e_j \in G.in\_edges(u_i)$ **do**
  if $0 \leq t_i - t_j - \delta_j \leq d$ **then**
    in_source $\leftarrow e_j$
  end if
end for
return in_source
end function

function otherwise(G, e_i, d)
out_source $\leftarrow \emptyset$, $u_i \leftarrow e_i$.source()
for all $e_j \in G.out\_edges(u_i)$ and $e_j \in G.in\_edges(v_i)$ **do**
  if $0 \leq t_j - t_i - \delta_i \leq d$ or $0 \leq t_i - t_j - \delta_j \leq d$ **then**
    otherwise $\leftarrow e_j$
  end if
end for
return otherwise
end function

4 Results

The algorithm presented in Section 3 was implemented in the Python programming language [16], using the NetworkX library [17] for processing graphs. Network data are visualized using the Gephi [18] visualization platform.

4.1 Network Data

The data used to demonstrate the ideas in this paper come from the Prosper Marketplace – a peer-to-peer lending system which allows borrowers and lenders to interact without intermediation from a bank. The Prosper Marketplace closed due to regulatory issues in November 2008, and was relaunched in July 2009. A nightly snapshot of data is released on the Prosper website, giving information about bids, listings, members, groups and loans.

In our previous work [19], an analysis of the network composed of members who interacted via loans is presented. This network is represented as a directed graph in which multiple edges between vertices are allowed, but self-loops aren't. The network was time-sliced into snapshots each of four months duration, with a sliding window of two months. In order to examine how groups of users might collaborate – perhaps with the intention of defrauding the system – we sought dense structures within the individual networks and plotted their occurrences over time. It was seen that there are significantly more structures composed of directed paths before the temporary closure of the Prosper system than after. These were captured in our extraction of fan-out-fan-in structures from a single source vertex to a single target vertex composed of paths of up to 6 hops.

This approach is subject to the same potential pitfalls discussed in Section 2. The discovery of paths is restricted to the confines of the time window, so a path spanning more than four months – or indeed a path beginning at the end of a time window – will not be found. Within a time window, the amortization of a loan occurs and the associated funds may be propagated through the network at any time, so all interactions are more loosely defined as candidates to be included in a subgraph. This does not allow for fine-grained control over the actual duration and time between each interaction that would allow time-respecting subgraphs to be defined. For example, if a different interaction duration or time between interactions is required, the network must be time-sliced accordingly.

We expect to see the same behaviour emerge as in our previous work, with larger time-respecting subgraphs identified before the closure. One month from before the temporary closure of Prosper and one from after were chosen for comparison. The same month was selected in both cases, so seasonal variability can be safely ignored. The pre-closure network of September 2006 contains 3 753 vertices and 26 588 edges. The post-closure network of September 2009 contains 4 714 vertices and 29 010 edges. The density of the networks are 1.9×10^{-3} and 1.3×10^{-3} respectively. Since these densities are so similar, they are unlikely to explain differences in the sizes of time-respecting subgraphs.

4.2 Time-respecting Subgraphs Identified

Prosper repayments are monthly, so in the formulation of time-respecting subgraphs d may take values from within the range 0 days to 30 days. This follows from the assumption that if a borrower intends to default on a loan, this will become apparent by the first repayment. The transfer of money is instantaneous, so the interaction time $\delta = 0$

As the value of d approaches 30 days (which would subsume all of September), the number of time-respecting subgraphs approaches the number of weakly connected components. Both the pre- and post-closure networks contain two weakly connected components – one of size two and the other containing the rest of the edges. Thus, for values of d greater than 0, the number of disjoint time-respecting subgraphs decreases quickly. Thus, the figures in this paper result from identifying time-respecting subgraphs in which $d = 0$ or $d = 5$.

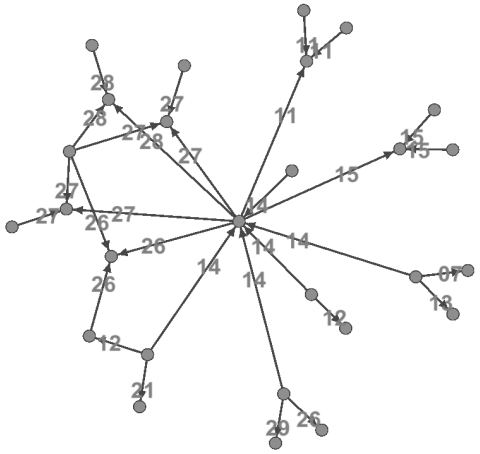


Fig. 3: Part of a time-respecting subgraph from September 2006, with $d = 5$.

Figure 3 illustrates a section of a subgraph in which interactions were allowed to occur within 5 days of each other. The individual we focus on received payments on the 14th of September, and lent money on the 11th, 15th, 26th, 27th and 28th. The out-edge on the 11th was admitted since under a comparison with any of the other out-edges from the same source it succeeds under case three of the definition. A case may be made for the inclusion of this behaviour (the individual began lending under the assumption that a loan would soon be received), or it could be excluded by making the definition stricter.

Figure 4 shows an excerpt from a subgraph from the pre-closure era. Interactions occurred on the 12th of the month. The ego mostly acted as a borrower, but made one bid on a loan in the capacity of a lender.

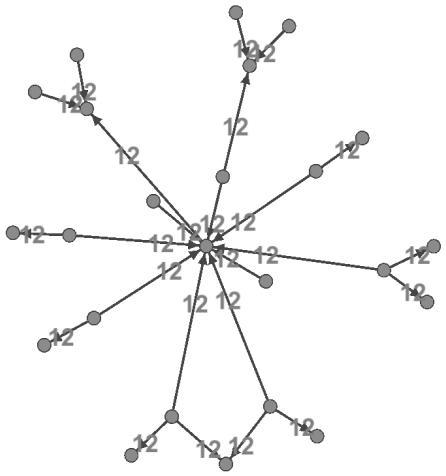


Fig. 4: Part of a time-respecting subgraph from September 2006, with $d = 0$.

Fig. 5: Part of a time-respecting subgraph from September 2009, with $d = 0$.

Figure 5 illustrates a section of a subgraph which occurred after the temporary closure of Prosper. The individual of interest lent once on the 30th and otherwise borrowed on that particular day. It is interesting to see that two vertices interacting with the ego are also connected, and on the same day.

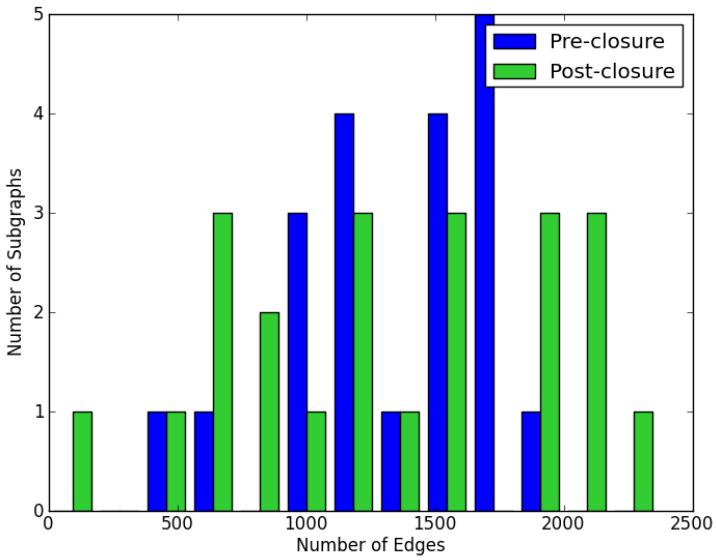


Fig. 6: Subgraph size distributions during the pre- and post-closure time frames.

Figure 6 presents a comparison of the number of time-respecting subgraphs before and after the temporary closure of Prosper when $d = 0$, the most limiting case. The range of edge counts on the x-axis begins at three, to avoid the display being skewed by individual edge pairs and singleton edges. In accordance with our expectations, the sizes of subgraphs identified before the closure exceed the sizes of those identified afterwards in the majority of cases, with the exception of very large subgraphs.

5 Conclusions and Future Work

In the many types of network imbued with temporal information, knowledge of the routes along which information can be sequentially passed leads to a greater understanding of the system under study. Presented in this paper is an algorithm to identify these routes, which we term time-respecting subgraphs. When applied to the Prosper data set, the algorithm reveals the difference in behaviour of users before and after regulatory tightening.

In the quest for anomalous structure, the filtering afforded by tighter temporal restrictions is valuable. If the network was viewed as static, such savings could not be made and important knowledge of sequential processes would not be available. The aggregated network approach – for example using a sliding

window of one week – would fail to capture flows of interaction exceeding this time-span.

However, a similar although more manageable problem exists with the choice of d, which defines the lifespan of a piece of information. In the case where $d = 0$, a new interaction must commence instantaneously after the end of the preceding interaction, yielding small subgraphs. If d is as large as the time difference between the first and last interaction in the entire network, the whole network may be included in a single time-respecting subgraph (assuming the required connectivity). Thus, the choice of d expresses the level of granularity required for the time-respecting subgraphs. Despite this, the representation we construct is more scalable in terms of memory than that composed of snapshots of a network at different time windows, which would require a copy of all vertices for each time window.

In our continuing study of temporal networks, we intend to write a more efficient, scalable version of the algorithm presented in this paper. This would allow us to quickly identify time-respecting subgraphs with a range of values for the necessary parameters. When applied to different types of data set, the results may give us greater insight into the actual meaning encoded in temporal networks.

References

1. Holme, P., Saramäki, J.: Temporal networks. CoRR **abs/1108.1780** (2011)
2. Pan, R.K., Saramäki, J.: Path lengths, correlations, and centrality in temporal networks. CoRR **abs/1101.5913v2** (2011)
3. Faloutsos, C., McCurley, K.S., Tomkins, A.: Connection subgraphs in social networks. In Proceedings of the Workshop on Link Analysis, Counterterrorism, and Privacy (in conj. with SIAM International Conference on Data Mining) (2004)
4. Moody, J.: The importance of relationship timing for diffusion. Social Forces **81** (2002) 25–56
5. Prosper Marketplace: Prosper (2012)
6. Krings, G., Karsai, M., Bernharsson, S., Blondel, V., Saramäki, J.: Effects of time window size and placement effects of time window size and placement on the structure of aggregated networks. EPJ Data Science **1**(4) (2012)
7. Kempe, D., Kleinberg, J., Kumar, A.: Connectivity and inference problems for temporal networks. Journal of Computer and System Sciences **76**(036113) (2002)
8. Nicosia, V., Tang, J., Musolesi, M., Russo, G., Mascolo, C., Latora, V.: Components in time-varying graphs. CoRR **abs/1106.2134** (2012)
9. Cheng, E., Grossman, J.W., Lipman, M.J.: Time-stamped graphs and their associated influence digraphs. Discrete Applied Mathematics **128** (2003) 317–335
10. Bearman, P., Moody, J., Stovel, K.: Chains of affection: The structure of adolescent romantic and sexual networks. American Journal of Sociology **110** (2004) 44–91
11. Riolo, C.S., Koopman, J.S., Chick, J.S.: Methods and measures for the description of epidemiological contact networks. Journal of Urban Health **78** (2001) 446–457
12. Zhao, Q., Tian, Y., He, Q., Oliver, N., Jin, R., Lee, W.C.: Communication motifs: A tool to characterize social communications. In Proceedings of the 19th ACM International Conference on Information and Knowledge Management (2010) 1645

13. Kivelä, M., Pan, R.K., Kaski, K., Kertész, J., Saramäki, J., Karsai, M.: Multiscale analysis of spreading in a large communication network. CoRR (2011)
14. Tang, J., Musolesi, M., Mascolo, C., Latora, V., Nidosia, V.: Analysing information flows and key mediators through temporal centrality metrics. Proceedings of the 3rd Workshop on Social Networks Systems 3 (2010)
15. Hopcroft, J., Tarjan, R.: Efficient algorithms for graph manipulation. Communications of the ACM **16**(6) (1973) 372–378
16. Python Software Foundation: Python (2012)
17. NetworkX Developers: Networkx (2010)
18. Gephi: Gephi. gephi.org (2012)
19. Redmond, U., Harrigan, M., Cunningham, P.: Mining dense structures to uncover anomalous behaviour in financial network data. In: Mining Ubiquitous and Social Environments. Volume 7472 of Lecture Notes in Artificial Intelligence. Springer-Verlag (2012 (to appear)) 60–76