DyNaK 2010
Dynamic Networks and Knowledge Discovery

Proceedings of the 1st Workshop on Dynamic Networks and Knowledge Discovery co-located with ECML PKDD

2010 Barcelona, Spain, September 24, 2010

Ruggero G. Pensa
Francesca Cordero
Céline Rouveirol
Rushed Kanawati
José Troyano
Paolo Rosso

Editors' address:

Ruggero G. Pensa
Dipartimento di Informatica - Università di Torino
C.so Svizzera 185, I-10149 Torino, Italy
pensa@di.unito.it

Francesca Cordero
Dipartimento di Informatica - Università di Torino
C.so Svizzera 185, I-10149 Torino, Italy
fcordero@di.unito.it

Céline Rouveirol
LIPN - UMR CNRS 7030
Institut Galilée - Université Paris-Nord
99, avenue Jean-Baptiste Clément
93430 Villetaneuse, France
celine.rouveirol@lipn.univ-paris13.fr

Rushed Kanawati
LIPN - UMR CNRS 7030
Institut Galilée - Université Paris-Nord
99, avenue Jean-Baptiste Clément
93430 Villetaneuse, France
rushed.kanawati@lipn.univ-paris13.fr

José Troyano
Universidad de Sevilla
Av. Reina Mercedes, s/n., 41012 Sevilla, Spain
troyano@us.es

Paolo Rosso
Universidad Politécnica de Valencia, DSIC
edificio 1F, Camino de Vera s/n, 46022 Valencia, Spain
prosso@dsic.upv.es

Foreword

Modeling and analyzing networks is a major emerging topic in different research areas, such as computational biology, social science, document retrieval, etc. By connecting objects, it is possible to obtain an intuitive and global view of the relationships between components of a complex system.

Nowadays, the scientific communities have access to huge volumes of network-structured data, such as social networks, gene/proteins/metabolic networks, sensor networks, peer-to-peer networks. Most often, these data are not only static, but they are collected at different time points. This dynamic view of the system allows the time component to play a key role in the comprehension of the evolutionary behavior of the network (evolution of the network structure and/or of flows within the system). Time can help to determine the real causal relationships within, for instance, gene activations, link creation, information flow.

Handling such data is a major challenge for current research in machine learning and data mining, and it has led to the development of recent innovative techniques that consider complex/multi-level networks, time-evolving graphs, heterogeneous information (nodes and links), and requires scalable algorithms that are able to manage huge and complex networks.

DyNaK workshop is motivated by the interest of providing a meeting point for scientists with different backgrounds that are interested in the study of large complex networks and the dynamic aspects of such networks. It includes contributions from both aspects of networks analysis: large real network analysis and modelling, and knowledge discovery within those networks. Even though each type of real complex networks has some peculiarities related to its specific domain, many aspects of the modeling and mining techniques for such networks are shareable. For instance, gene networks and social networks share a common architecture (scale-free), and involve similar data mining and machine learning methods: module/community extraction, hub single-out, information-flow analysis, missing link detection and link prediction.

DyNaK also host a special session on Sentiment Analysis and Opinion Mining. Every day, millions of people write their opinions about any issue in social media, such as social news sites, review sites, and blogs. The distillation of knowledge from this huge amount of unstructured information is a challenging task. Sentiment Analysis and Opinion Mining are two areas related to Natural Language Processing and Text Mining that deal with the identification of opinions and attitudes in natural language texts. The Opinion Mining session of DyNaK includes results from academics and practitioners in the task of extracting knowledge from user generated contents.

We received 18 submissions: 7 were accepted as long presentations, and 2 as short presentations. In addition to the technical papers, the program also includes three invited talks by Tanya Berger-Wolf (University of Illinois, USA), Stefan Kramer (Technische Universität München, Germany) and Carlos Rodrìguez (Research Center, Barcelona-Media, Spain), and an industrial keynote by Enrico Bucci (BioDigitalValley Srl, Italy).

We would like to thank the Program Committee members without whom the preparation of this program would not have been possible. Many thanks to Sébastien Guérif, for his help when editing these proceedings.

Our gratitude also goes to BioDigitalValley Srl and the Computer Science Lab of the Paris-Nord University, that co-supported our workshop. Finally, we are thankful to the Computer Science Department of the University of Torino, and the SysBioM Center, which supported our activities.

Ruggero G. Pensa
Francesca Cordero
Céline Rouveirol
Rushed Kanawati
José Troyano
Paolo Rosso

Program Committee

- Riccardo Bellazzi, University of Pavia, Italy
- Guillaume Beslon, INSA-Lyon, France
- Bettina Berendt, Katholieke Universiteit Leuven, Belgium
- Tanya Berger-Wolf, University of Illinois, USA
- Karsten M. Borgwardt, MPI, Tübingen, Germany
- Jean-François Boulicaut, INSA-Lyon, France
- Raffaele Calogero, University of Torino, Italy
- Iván Cantador, Universidad Autónoma de Madrid, Spain.
- Francisco M. Carrero, Universidad Europea de Madrid, Spain
- José C. Cortizo, Universidad Europea de Madrid, Spain
- Diego Di Bernardo, TIGEM, Italy
- Mohamed Elati, University of Evry, France
- Paolo Frasconi, University of Firenze, Italy
- Lise Getoor, University of Maryland, USA
- Dino Ienco, University of Torino, Italy
- Tamara G. Kolda, Sandia National Laboratories, USA
- Stefan Kramer, Technische Universität München, Germany
- Tao Li, Florida International University, USA
- Pietro Liò, University of Cambridge, UK
- Huan Liu, Arizona State University, USA
- Eric Yu-En Lu, University of Cambridge, UK
- Sara C. Madeira, INESC-ID/IST, Portugal
- Rosa Meo, University of Torino, Italy
- Tsuyoshi Murata, Tokyo Institute of Technology, Japan
- Mirco Nanni, ISTI-CNR, Italy
- Arlindo Oliveira, INESC-ID, Portugal
- Andrea Passerini, University of Trento, Italy
- Lorenza Saitta, University of Piemonte Orientale, Italy
- Rossano Schifanella, University of Torino, Italy
- Einoshin Suzuki, Kyushu University, Japan
- Hanghang Tong, Carnegie Mellon University, USA

Organizing Committee

- Ruggero G. Pensa, University of Torino, Italy
- Francesca Cordero, University of Torino, Italy
- Céline Rouveirol, University of Paris Nord, France
- Rushed Kanawati, University of Paris Nord, France
- José A. Troyano, University of Sevilla, Spain
- Paolo Rosso, Technical University of Valencia, Spain

Contents

Finding structure in dynamic networks and what it means for zebras

Tanya Berger-Wolf

University of Illinois, USA

Abstract

Social creatures interact in diverse ways: forming groups, mating, sending emails, and sharing ideas. Some of the interactions are accidental while others are a consequence of the underlying explicit or implicit social structures. One of the most important questions in sociology is the identification of such structures, which are variously viewed as communities, hierarchies, or "social profiles".

In analyzing social networks, one property has largely been ignored until recently: interactions and their nature change over time. The notion of "structure" is intricately linked with the dynamics of social interactions. On one hand, it is in longitudinal data that the emergence of structures and the laws governing their development can be observed and inferred. On the other hand, the existence of such structures that constrain social interactions is what allows us to predict the behavior and nature of dynamic networks. The necessity to delve into the dynamic aspects of networking behavior may be clear, yet it would not be feasible without the data to support such explicitly dynamic analysis. Rapidly growing electronic networks, such as emails, the Web, blogs, and friendship sites, as well as mobile sensor networks on cars, humans, and animals, provide an abundance of dynamic social network data that for the first time allow the temporal component to be explicitly addressed in network analysis.

I will present several examples of computational approaches we have developed to infer structure in dynamic networks and show applications of this analysis to population biology, from humans to zebras.

Learning Real-Time Automata from Multi-Attribute Event Logs

Stefan Kramer

Technische Universität München, Germany

Abstract

Network structures often arise as descriptions of complex temporal phenomena in science and industry. Popular representation formalisms include Petri nets and (timed) automata. In process mining, the induction of Petri net models from event logs has been studied extensively. Less attention, however, has been paid to the induction of (timed) automata outside the field of grammatical inference. In the talk, I will present work on the induction of timed automata and show how they can be learned from multi-attribute event logs. I will present the learning method in some detail and give examples of network inference from synthetic, medical and biological data.

Protein-centered biological networks by automatic caption analysis

Enrico M. Bucci

BioDigitalValley Srl, Italy

Abstract

In former years, a lot of attention has been paid to the retrieval of meaningful biological information connecting proteins and genes, i.e. relationships between different players in the cascade of molecular events regulating the physiology and pathology of cells, tissues and eventually organisms. The main goal is to develop genes/proteins connection models able to explain complex biological phenomena in terms of emerging properties of large, structured networks, whose topology and detailed structure account at least in part for these properties. This implies the use of experimental methods able to collect information on a large number of different proteins under different conditions, and then properly connecting the data to the results obtained all over the world, so to get a coherent picture in a larger frame. In particular, to encompass a larger body of information and to figure out how some experimental study fits to the accumulated knowledge, methods are required to retrieve the available data on all proteins involved in the study (the target proteins), as well as on all proteins, which are connected by some piece of information to the targets. To this aim, a method consists in parsing automatically the scientific literature, retrieving co-occurring names of proteins, genes or other kinds of molecules and attempting to identify some terms which qualifies the relationship between the identified proteins. This task is a non trivial one, giving the ambiguity in gene/protein nomenclature (which affects both precision and recall of the relevant data), and the strong dependence of the type of relationship on the context at multiple levels. Most of the available methods parse only the abstracts of the scientific literature; however, the information contained in the abstracts is often incomplete, due to the fact that only those genes/proteins which are in the main scope of the paper are discussed, while often data on a number of other proteins are contained elsewhere.

In an attempt to overcome these limitations, we focussed on the analysis of the figure captions contained in the scientific literature. The captions of a paper refer in most cases to the experiments described in the paper, and thus contain an enriched amounts of data describing the biology of different proteins, including the relationship between them. Moreover, since terms referring to gene/proteins and other terms related to experimental methodologies are simultaneously present in a reduced textual space, it is possible to identify groups of proteins studied with a certain experimental technique; by properly filtering for a specific technique, is possible to characterize the type of relationship between the proteins. For example, proteins co-occurring in a caption describing

a double-hybrid experiment are most likely binding partners, while proteins co-occurring in a caption describing a 2D-gel experiments are probably co-expressed in a given condition/biological sample.

We thus developed Protein Quest, a tool which automatically and efficiently parse both the abstract and the captions of scientific paper in a pdf document. Results obtained from more than 2.000.000 free, full-text papers will be discussed, with reference to the topological characterization of the obtained cooccurrence networks and to the dependence of their topology from different query strategies; moreover, some specific, disease-oriented networks and predictions will be presented.

Discovering Inter-Dimensional Rules in Dynamic Graphs

Kim-Ngan T. Nguyen[1], Loïc Cerf[1], Marc Plantevit[2], and Jean-François Boulicaut[1]

[1] Université de Lyon, CNRS, INRIA
INSA-Lyon, LIRIS Combining, UMR5205, F-69621, France
[2] Université de Lyon, CNRS, INRIA
Université Lyon 1, LIRIS Combining, UMR5205, F-69622, France

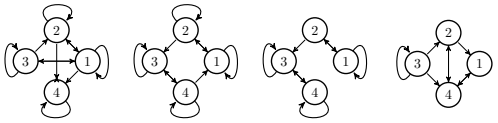
Abstract. Data mining methods that exploit graph/network have become quite popular and a timely challenge is to consider the discovery of dynamic properties in evolving graphs or networks. In this paper, we consider the dynamic oriented graphs that can be encoded as n-ary relations with $n \geq 3$ such that we have a least 3 dimensions: the dimensions of departure (tail) and arrival (head) vertices plus the time dimension. In other terms, it encodes the sequence of adjacency matrices of the graph. In such datasets, we propose a new semantics for inter-dimensional rules in dynamic graphs. We define rules that may involve subsets of any dimensions in their antecedents and their consequents and we propose the new objective interestingness measure called the exclusive confidence. We introduce a first algorithm for computing such inter-dimensional rules and we illustrate the added-value of exclusive confidence for supporting the discovery of relevant rules from a real-life dynamic graph.

1 Introduction

Graph mining is a popular topic. Many researchers have considered pattern discovery from large collections of graphs while others focus the analysis of one large graph or network. In the latter case, we observe two complementary directions of research. On one hand, global properties of graphs are studied (e. g., power-law distribution of node degrees or diameters). On the other hand, it is possible to use data mining algorithms to identify local patterns in the graphs (e. g., frequent subgraphs, clique patterns). Such local techniques can indeed benefit from the huge research effort on 0/1 data analysis, i. e., a graphs is seen as particular 0/1 table (the two involved domains being identical): its adjacency matrix.

In this paper, we investigate local pattern discovery from dynamic directed graphs, i. e., from of collection of static directed graphs that all share the same set of uniquely identified vertices. For instance, Fig. 1 depicts a dynamic directed graph involving four nodes. Four snapshots of this graph are available. The dynamic graph can be represented as the sequence of its adjacency matrices underneath. It describes the relationship between the tail vertices in $D^1 = \{d_1, d_2, d_3, d_4\}$ and the head vertices in $D^2 = \{a_1, a_2, a_3, a_4\}$ at the timestamps

in $D^3 = \{t_1, t_2, t_3, t_4\}$. Every '1', in the adjacency matrices is at the intersection of three elements $(d_i, a_j, t_k) \in D^1 \times D^2 \times D^3$, which indicate a directed edge from d_i to a_j at time t_k. Therefore at least three dimensions are necessary to encode a dynamic graph, which can be seen as a ternary relation (the one depicted in Fig. 1 is called $\mathcal{R}_E$). However, more dimensions may be used, for instance to encode information on edges and/or time aspects with different granularity.



	a_1	a_2	a_3	a_4	a_1	a_2	a_3	a_4	a_1	a_2	a_3	a_4	a_1	a_2	a_3	a_4
d_1	1	1	1	1	1	1			1	1						1
d_2	1	1		1	1	1	1		1		1		1			1
d_3	1	1	1				1	1			1	1		1	1	1
d_4				1	1		1	1			1	1	1	1		
	t_1				t_2				t_3				t_4			

Fig. 1: The dynamic graph $\mathcal{R}_E \subseteq \{d_1, d_2, d_3, d_4\} \times \{a_1, a_2, a_3, a_4\} \times \{t_1, t_2, t_3, t_4\}$.

Studying descriptive rule mining from dynamic graphs is a rather new research topic and most of previous work impose severe restrictions on the form of the rules. The key contribution of this paper is the proposal of a quite general form of rules. These rules may involve any subset of dimensions in both the left-hand side and the right-hand side. In particular, the temporal dimensions can either explicitly appear in the rules or be used to measure the importance of the rules (i. e., the number of timestamps where the rule holds). Taking into account these different ways is complementary. It provides relevant patterns describing the evolution of a dynamic graph at a local level. Two examples of inter-dimensional rules that we want to extract are given in Fig. 2. Fig. 2a depicts a rule that is preserved at several timestamps. It intuitively means that if, at a time, the edges from vertices 2, 3 and 4 have the same heads then these heads are exclusively vertex 3. Rule in Fig. 2b means that if there are pairs of edges whose tails are nodes 3 and 4 and whose heads are the same vertex then it mainly occurs at times t_2 and t_3. To express the a priori relevancy of such rule, we use a straightforward extension of the classical frequency measure and an original extension of the confidence measure, the so-called *exclusive confidence*. The second contribution of this paper deals with the design of an algorithm that computes the a priori interesting rules. It exploits the principles (typically the enumeration strategy) of [7], i. e., the state-of-the-art algorithm for exploring the search space of multi-dimensional associations.

In Sect. 2, we provide the needed definitions to build the new pattern domain of inter-dimensional rules. Then, in Sect. 3, we define such rules and the exclusive

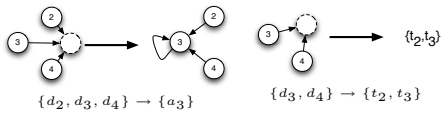


Fig. 2: Example of rules.

confidence semantics. Sect. 4 introduces the first algorithm that computes a priori interesting rules from a dynamic graph. Sect. 5 deals with the empirical validation and various experiments on a real-life dynamic graph. Sect. 6 discusses related work and, finally, Sect. 7 briefly concludes.

2 Preliminary Definitions

Given n finite domains $\mathcal{D} = \{D^1, \ldots, D^n\}$ and an n-ary relation $\mathcal{R} \subseteq \times_{i=1..n} D^i$, the patterns of interest only involve some of the domains $\mathcal{D}' \subseteq \mathcal{D}$. E. g., the analyst may want to focus on subgraph patterns ($\mathcal{D}' = \{D_1, D_2\}$ in $\mathcal{R}_E$). She may, instead, want to discover pattern involving temporal dimensions. Without loss of generality, the dimensions are assumed ordered such that $\mathcal{D}' = \{D^1, \ldots, D^{|\mathcal{D}'|}\}$. We now formally define an *association* on $\mathcal{D}'$.

Definition 1 (Association). *$\forall \mathcal{D}' = \{D^1, \ldots, D^{|\mathcal{D}'|}\} \subseteq \mathcal{D}$, $\times_{i=1..|\mathcal{D}'|} X^i$ is an association on $\mathcal{D}'$ iff $\forall i = 1..|\mathcal{D}'|$, $X^i \neq \emptyset \wedge X^i \subseteq D^i$.*

$\times_{D^i \in \mathcal{D} \setminus \mathcal{D}'} D^i$ is called *support domain*. The support of an association generalizes that of an *itemset* in a binary relation ($n = 2$ and $|\mathcal{D}'| = 1$) [1]. Its formal definition uses the concatenation operator, denoted '$\cdot$'. E. g., $(d_2, a_3) \cdot (t_1) = (d_2, a_3, t_1)$.

Definition 2 (Support s). *$\forall \mathcal{D}' \subseteq \mathcal{D}$, let X an association on $\mathcal{D}'$. Its support, denoted $s(X)$, is $s(X) = \{u \in \times_{D^i \in \mathcal{D} \setminus \mathcal{D}'} D^i \mid \forall x \in X,\ x \cdot u \in \mathcal{R}\}$.*

The following definitions will ease the exposition of this paper.

Definition 3 (Projection π). *$\forall \mathcal{D}' = \{D^1, \ldots, D^{|\mathcal{D}'|}\} \subseteq \mathcal{D}$, let $X = X^1 \times \cdots \times X^{|\mathcal{D}'|}$ an association on $\mathcal{D}'$. $\forall D^i \in \mathcal{D}$, $\pi_{D^i}(X)$ is X^i if $D^i \in \mathcal{D}'$, $\emptyset$ otherwise.*

Definition 4 (Union $\sqcup$). *$\forall \mathcal{D}_X \subseteq \mathcal{D}$ and $\forall \mathcal{D}_Y \subseteq \mathcal{D}$, let X an association on $\mathcal{D}_X$ and Y an association on $\mathcal{D}_Y$. $X \sqcup Y$ is the association on $\mathcal{D}_X \cup \mathcal{D}_Y$ for which $\forall D^i \in \mathcal{D}$, $\pi_{D^i}(X \sqcup Y) = \pi_{D^i}(X) \cup \pi_{D^i}(Y)$.*

Definition 5 (Complement $\setminus$). *$\forall \mathcal{D}_X \subseteq \mathcal{D}$ and $\forall \mathcal{D}_Y \subseteq \mathcal{D}$, let X an association on $\mathcal{D}_X$ and Y an association on $\mathcal{D}_Y$. $Y \setminus X$ is the association on $\{D^i \in \mathcal{D}_Y \mid \pi_{D^i}(Y) \not\subseteq \pi_{D^i}(X)\}$ for which $\forall D^i \in \mathcal{D}$, $\pi_{D^i}(Y \setminus X) = \pi_{D^i}(Y) \setminus \pi_{D^i}(X)$.*

In $\mathcal{R}_E$, depicted in Fig. 1, $\{d_1, d_2\} \times \{a_1, a_2\}$ is an association on $\{D^1, D^2\}$, whereas $\{a_1, a_2\}$ is not because $\pi_{D^1}(\{a_1, a_2\}) = \emptyset$. It is an association on $\{D^2\}$. Their respective supports are $s(\{d_1, d_2\} \times \{a_1, a_2\}) = \{t_1, t_2\}$ and $s(\{a_1, a_2\}) = \{(d_1, t_1), (d_1, t_2), (d_1, t_3), (d_2, t_1), (d_2, t_2), (d_3, t_1), (d_4, t_4)\}$.

3 Inter-Dimensional Rules in Dynamic Graph

Given a dynamic graph, encoded as an n-ary relation $\mathcal{R}$ on $\mathcal{D}$, the analyst chooses the domains $\mathcal{D}_X \subseteq \mathcal{D}$ and $\mathcal{D}_Y \subseteq \mathcal{D}$ at, respectively, the left-hand side and the right-hand side of the rules to discover. E. g., to list rules involving tail vertices at their antecedents and timestamps at their consequents, $\mathcal{D}_X$ only contains one dimension of the relation (the tail vertices) and so does $\mathcal{D}_Y$ (the timestamps). Notice that $\mathcal{D}_X \cap \mathcal{D}_Y$ must be empty. An inter-dimensional rule on $(\mathcal{D}_X, \mathcal{D}_Y)$ is a couple of associations[3]. The first one on $\mathcal{D}_X$, the second one on $\mathcal{D}_Y$.

Definition 6 (Inter-dimensional rule). *$\forall \mathcal{D}_X \subseteq \mathcal{D}$, $\forall \mathcal{D}_Y \subseteq \mathcal{D}$, $X \to Y$ is an inter-dimensional rule on $(\mathcal{D}_X, \mathcal{D}_Y)$ iff X is an association on $\mathcal{D}_X$, Y is an association on $\mathcal{D}_Y$ and $\mathcal{D}_X \cap \mathcal{D}_Y = \emptyset$.*

In $\mathcal{R}_E$, $\{d_3\} \to \{a_3, a_4\}$ is an inter-dimensional rule on $(\{D^1\}, \{D^2\})$. The rule $\{d_3\} \times \{a_3, a_4\} \to \{d_4\}$ is not an inter-dimensional rule because elements in D^1 appear both at its left-hand side and at its right-hand side.

A rule is frequent if many "objects" verifies it. These objects are elements of a *support domain* for the rule, which is, in fact, $\times_{D^i \in \mathcal{D} \setminus (\mathcal{D}_X \cup \mathcal{D}_Y)} D^i$, i. e., that of the association (on $\mathcal{D}_X \cup \mathcal{D}_Y$) union of its antecedent and its consequent. The rule can be trusted, i. e., has a large enough confidence, if there is a high conditional probability to observe the consequent when the antecedent holds. In the context of inter-dimensional rules in dynamic graphs, a natural definition of the frequency exists. On the contrary, it is hard to define a confidence measure.

The (relative) frequency of an inter-dimensional rule in a dynamic graph is, in the support domain, the proportion of elements in the support of the union of its antecedent and its consequent.

Definition 7 (Frequency). *$\forall \mathcal{D}_X \subseteq \mathcal{D}$, $\forall \mathcal{D}_Y \subseteq \mathcal{D}$, the frequency of an inter-dimensional rule $X \to Y$ on $(\mathcal{D}_X, \mathcal{D}_Y)$ is $f(X \to Y) = \frac{|s(X \sqcup Y)|}{|\times_{D^i \in \mathcal{D} \setminus (\mathcal{D}_X \cup \mathcal{D}_Y)} D^i|}$.*

In $\mathcal{R}_E$, recursively applying Definitions 7, 4 and 2 gives $f(\{d_3\} \to \{a_3, a_4\}) = \frac{|s(\{d_3\} \times \{a_3, a_4\})|}{|D^3|} = \frac{|\{t_2, t_3, t_4\}|}{|\{t_1, t_2, t_3, t_4\}|} = \frac{3}{4}$.

Is it sensible to directly generalize the confidence measure of association rules in binary relations to n-ary relations? Doing so, the confidence of a rule $X \to Y$ would be $\frac{|s(X \sqcup Y)|}{|s(X)|}$. Unfortunately, this semantics is not satisfactory. Indeed, $s(X \sqcup Y)$ and $s(X)$ are disjoint sets and the ratio of their cardinalities does not make any sense. For instance, in $\mathcal{R}_E$, consider the rule $\{d_3\} \to \{a_3, a_4\}$. We have $s(\{d_3\} \times \{a_3, a_4\}) = \{t_2, t_3, t_4\}$ (i. e., a set of timestamps) while $s(\{d_3\}) = \{(a_1, t_1), (a_2, t_1), (a_3, t_1), (a_3, t_2), \dots\}$ (i. e., a set of couples (head vertices,timestamps)). However, it is possible to introduce a factor such that $|s(X)|$ and $|s(X \sqcup Y)|$ become comparable. The idea is to multiply $|s(X \sqcup Y)|$ by the cardinalities of its projections in the domains in $\mathcal{D}_Y$.

[3] The term "inter-dimensional association rule" often means, in the literature, a rule with *one* element per dimension. Our definition is more general.

Definition 8 (Confidence). *$\forall \mathcal{D}_X \subseteq \mathcal{D}$, $\forall \mathcal{D}_Y \subseteq \mathcal{D}$, the (exclusive) confidence of an inter-dimensional rule $X \rightarrow Y$ on $(\mathcal{D}_X, \mathcal{D}_Y)$ is $c(X \rightarrow Y) = \frac{|s(X \sqcup Y)| \times |\times_{D^i \in \mathcal{D}_Y} \pi_{D^i}(Y)|}{|s(X)|}$.*

Roughly speaking, the remedial factor, applied to $|s(X \sqcup Y)|$, allows to count the elements in $s(X \sqcup Y)$ "in the same way at the numerator and at the denominator of the fraction". For example, consider the rule $\{d_3\} \rightarrow \{a_3, a_4\}$ in $\mathcal{R}_E$, its exclusive confidence is $c(\{d_3\} \rightarrow \{a_3, a_4\}) = \frac{|s(\{d_3\} \times \{a_3, a_4\})| \times |\{a_3, a_4\}|}{|s(\{d_3\})|} = \frac{6}{10}$. Fig. 3 depicts, at every timestamp, the dynamic graph in Fig. 1 but it only keeps the ten edges with the vertex 3 as a tail. This number, "10", is found at the denominator of the fraction to compute the confidence. At its numerator, "6" actually is the count of those, among these 10 edges, that go to the vertices 3 and 4 at the same timestamp. They are thick in Fig. 3. At time t_1, there is an edge from d_3 to a_3 but there is no edge from d_3 to a_4 at this time. This "lowers" the confidence of the rule because a_4 is at its consequent too. At time t_4, there is an edge from d_3 to a_2. This also "lowers" the confidence in the fact that if d_3 is the tail of an edge then its head is either a_3 or a_4 (and not another vertex). That is why, this semantics of the confidence is said "exclusive". If $c(\{d_3\} \rightarrow \{a_3, a_4\})$ was 1, i. e., the maximal possible value, then, in every snapshot of the graph where the vertex 3 has a non-null output degree, it would *always* have two outgoing edges that would bind it with the vertex 3 and 4. *Any* other edge, with the vertex 3 as its tail, "lowers" the confidence.

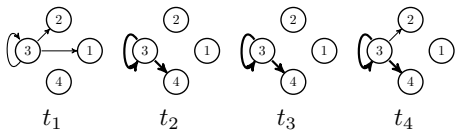


Fig. 3: Computing the confidence of $\{d_3\} \rightarrow \{a_3, a_4\}$.

Notice that the same speech applies to inter-dimensional rules involving the temporal dimension. E. g., Fig. 3 could illustrate, as is, the computation of $c(\{d_3\} \rightarrow \{t_2, t_3, t_4\})$, hence the same result $\frac{6}{10}$. This time however, the tick edges must be understood as those shared by the snapshots of the dynamic graph at t_2, t_3 and t_4 ("edgewise and" operation between the three graphs).

4 Computing Rules

Given an n-ary relation $\mathcal{R} \subseteq \times_{D^i \in \mathcal{D}} D^i$ and the parameters $(\mathcal{D}_X, \mathcal{D}_Y)$ (subsets of $\mathcal{D}$ and such that $\mathcal{D}_X \cap \mathcal{D}_Y = \emptyset$), $\mu \in [0, 1]$ and $\beta \in [0, 1]$, the *a priori* interesting inter-dimensional rules $X \rightarrow Y$ are such that (i) X is an association on $\mathcal{D}_X$, (ii) Y is an association on $\mathcal{D}_Y$, (iii) $f(X \rightarrow Y) \geq \mu$ and (iv) $c(X \rightarrow Y) \geq \beta$.

Our method, namely GEAR, first rewrites the relation by combining the components which are neither in $\mathcal{D}_X$ nor in $\mathcal{D}_Y$. In other terms, it builds the

support domain $D^{\text{supp}} = \times_{D^i \in \mathcal{D} \setminus (\mathcal{D}_X \cup \mathcal{D}_Y)} D^i$. The resulting relation, $\mathcal{R}_A$ is defined on the dimensions $\mathcal{D}_A = \mathcal{D}_X \cup \mathcal{D}_Y \cup \{D^{\text{support}}\}$. Then, GEAR extracts, in $\mathcal{R}_A$, every association A on $\mathcal{D}_X \cup \mathcal{D}_Y$ satisfying $\frac{|s(A)|}{|D^{\text{supp}}|} \geq \mu$. It entails that $\sqcup_{D^i \in \mathcal{D}_X} \pi_{D^i}(A) \rightarrow \sqcup_{D^i \in \mathcal{D}_Y} \pi_{D^i}(A)$ is a frequent inter-dimensional rule (and reciprocally, hence the completeness). Its exclusive confidence is finally computed. If it exceeds β, the rule is output.

The actual extraction of every frequent association A (associated with its support $A^{\text{supp}} \subseteq D^{\text{supp}}$), in $\mathcal{R}_A$, is now briefly detailed. A constraint-based approach is adopted, i. e., the problem is rewritten in terms of constraints and the patterns satisfying them all are the frequent associations. Here are the constraints:

- $\mathcal{C}_{\text{on-}(\mathcal{D}_X \cup \mathcal{D}_Y)}(A \sqcup A^{\text{supp}}) \equiv \forall D^i \in \mathcal{D}_X \cup \mathcal{D}_Y,\ \pi_{D^i}(A) \neq \emptyset$;
- $\mathcal{C}_{\text{connected}}(A \sqcup A^{\text{supp}}) \equiv A \sqcup A^{\text{supp}} \subseteq \mathcal{R}_A$;
- $\mathcal{C}_{\text{entire-supp}}(A \sqcup A^{\text{supp}}) \equiv A^{\text{supp}} = s(A)$;
- $\mathcal{C}_{\lceil \mu \times |D^{\text{supp}}| \rceil\text{-freq}}(A \sqcup A^{\text{supp}}) \equiv |A^{\text{supp}}| \geq \lceil \mu \times |D^{\text{supp}}| \rceil$.

Thanks to the last constraint, the frequency of the rule $\sqcup_{D^i \in \mathcal{D}_X} \pi_{D^i}(A) \rightarrow \sqcup_{D^i \in \mathcal{D}_Y} \pi_{D^i}(A)$ must reach or exceed μ. Indeed, $\frac{|s(A)|}{|D^{\text{supp}}|} \geq \mu$ is equivalent to $|s(A)| \geq \lceil \mu \times |D^{\text{supp}}| \rceil$ and, because the third constraint ($A^{\text{supp}} = s(A)$) must be satisfied as well, it is equivalent to $|A^{\text{supp}}| \geq \lceil \mu \times |D^{\text{supp}}| \rceil$. The third constraint, $\mathcal{C}_{\text{entire-supp}}$, forces a "closed" support. Indeed, by definition of the support (Definition 2), adding an element to A^{supp} ($= s(A)$) necessarily violates $\mathcal{C}_{\text{connected}}$. Thus, $\mathcal{C}_{\text{entire-supp}}(A \sqcup A^{\text{supp}})$ is equivalent to $\forall t \in D^{\text{supp}} \setminus A^{\text{supp}},\ A \sqcup \{t\} \not\subseteq \mathcal{R}_A$.

The algorithm traverses the search space by recursively partitioning it into two complementary parts ("divide and conquer"). In this way, a binary tree represents the performed enumeration. At every node of this tree, two associations, namely U and V, are updated. U is the smallest association that may be discovered in the enumeration sub-tree rooted by the node, whereas $U \sqcup V$ is the largest. That is why GEAR is initially called with $U = \emptyset$ and $V = \times_{D^i \in \mathcal{D}_A} D^i$. At every non-terminal node, an element e is chosen in $\cup_{D^i \in \mathcal{D}_A} \pi_{D^i}(V)$. In the enumeration sub-tree that derives from the first child, e is present in every U association (i. e., e is "moved" from V to U). In the enumeration sub-tree that derives from the second child, e is absent from every U association (i. e., e is "removed" from V). There are two reasons for an enumeration node to be a leaf of the enumeration tree. The first reason is that at least one of the four constraints is guaranteed to be violated by every U association in the sub-tree that would derive from the node. It happens when:

- $\exists D^i \in \mathcal{D}_X \cup \mathcal{D}_Y$ such that $\pi_{D^i}(U \sqcup V) = \emptyset$ ($\mathcal{C}_{\text{on-}(\mathcal{D}_X \cup \mathcal{D}_Y)}$ is violated);
- $\forall D^i \in \mathcal{D}_A, \pi_{D^i}(U) \neq \emptyset \wedge U \not\subseteq \mathcal{R}_A$ ($\mathcal{C}_{\text{connected}}$ is violated);
- $\exists t \in D^{\text{supp}} \setminus \pi_{D^{\text{supp}}}(U \sqcup V)$ such that $\Big((U \sqcup V) \setminus \pi_{D^{\text{supp}}}(U \sqcup V)\Big) \sqcup \{t\} \subseteq \mathcal{R}_A$ ($\mathcal{C}_{\text{entire-supp}}$ is violated);
- $|\pi_{D^{\text{supp}}}(U \sqcup V)| < \lceil \mu \times |D^{\text{supp}}| \rceil$ ($\mathcal{C}_{\lceil \mu \times |D^{\text{supp}}| \rceil\text{-freq}}$ is violated).

The proofs of these pruning properties are based on generalizations of monotone or anti-monotone properties that the four constraints have. The constraint

$\mathcal{C}_{\text{connected}}$ is monotone, i. e., if an association X violates the constraints then every larger association violates it as well. Since U is the smallest association in the sub-tree, $\neg\mathcal{C}_{\text{connected}}(U)$ is a safe pruning criterion. Dually, the three other constraints are anti-monotone, i. e., if an association X violates one of them then every smaller association violates it as well. That is why, to potentially prune the sub-tree rooted by the current enumeration node, their variables are replaced by the largest association in it: $U \sqcup V$. The second reason for an enumeration node to be a leaf is the actual discovery of a frequent association U. It happens when there is no more element to enumerate, i. e., when $V = \emptyset$.

An improved enumeration strategy avoids the generation of the nodes that violate $\mathcal{C}_{\text{connected}}$. To do so, in every first child (where an element e is "moved" to U), every element in $\cup_{D^i \in \mathcal{D}_A} \pi_{D^i}(V)$ that would violate $\mathcal{C}_{\text{connected}}$ if added to $U \sqcup \{e\}$ is "removed" from V. Algorithm 1 sums up the extraction of every frequent inter-dimensional association rules with high enough confidences. Other performance improvements (e. g., pertaining to the enforcement of $\mathcal{C}_{\text{entire-supp}}$) were implemented. They actually are analog to what is done in [7] for the extraction of closed patterns in n-ary relations. The **Choose** function is that of [7] too. Another useful feature, inherited from [7], is the ability to additionally and efficiently enforce any piecewise (anti)-monotone constraint the associations must satisfy. In some of the following experiments, the constraint $\mathcal{C}_{(\alpha^1,\ldots,\alpha^{|\mathcal{D}_X \cup \mathcal{D}_Y|})\text{-min-sizes}}$ (where $(\alpha^1, \ldots, \alpha^{|\mathcal{D}_X \cup \mathcal{D}_Y|}) \in \mathbb{N}^{|\mathcal{D}_X \cup \mathcal{D}_Y|}$) will be used:

$$\mathcal{C}_{(\alpha^1,\ldots,\alpha^{|\mathcal{D}_X \cup \mathcal{D}_Y|})\text{-min-sizes}}(A \sqcup A^{\text{supp}}) \equiv \forall D^i \in \mathcal{D}_X \cup \mathcal{D}_Y, |\pi_{D^i}(A)| \geq \alpha^i \ .$$

Algorithm 1: Algorithm GEAR.

Input: (U, V)
Output: Every *a priori* interesting association rule involving every element in $\cup_{D^i \in \mathcal{D}_X \cup \mathcal{D}_Y} \pi_{D^i}(U)$ and possibly some elements in $\cup_{D^i \in \mathcal{D}_X \cup \mathcal{D}_Y} \pi_{D^i}(V)$

if $\mathcal{C}_{on\text{-}(\mathcal{D}_X \cup \mathcal{D}_Y)}(U \sqcup V) \wedge \mathcal{C}_{entire\text{-}supp}(U \sqcup V) \wedge \mathcal{C}_{\lceil \mu \times |D^{supp}| \rceil\text{-}freq}(U \sqcup V)$ **then**
  if $V = \emptyset$ **then**
    if $c(\sqcup_{D^i \in \mathcal{D}_X} \pi_{D^i}(U) \rightarrow \sqcup_{D^i \in \mathcal{D}_Y} \pi_{D^i}(U)) \geq \beta$ **then**
      Output $\sqcup_{D^i \in \mathcal{D}_X} \pi_{D^i}(U) \rightarrow \sqcup_{D^i \in \mathcal{D}_Y} \pi_{D^i}(U)$;
    else
      Choose $e \in \cup_{D^i \in \mathcal{D}_A} \pi_{D^i}(V)$;
      GEAR($U \sqcup \{e\}$,($V \setminus \{e\}$) $\setminus$ ($f \in$ $\pi_{D^i}(V) \mid \neg C_{\text{connected}}(U \sqcup \{e\} \sqcup \{f\}))_{D^i \in \mathcal{D}_A}$);
      GEAR(U, $V \setminus \{e\}$);

5 Experimental Study

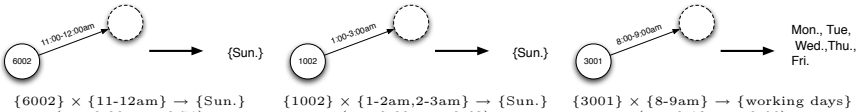
GEAR was implemented in C++ and compiled with GCC 4.2.4. This section reports experiments, which were performed on a GNU/Linux™ system equipped with an Intel® Pentium® 4 processor cadenced at 3 GHz and 1 GB of RAM.

`Vélo'v` is a bicycle rental service run by the urban community of Lyon, France. 327 `Vélo'v` stations are spread over Lyon and its surrounding area. At any of these stations, the users can take a bicycle and bring it to any other station. Whenever a bicycle is rented or returned, this event is logged. Logs represent more than 13.1 million rides along 30 months. This dataset is seen as a dynamic directed graph evolving into two temporal dimension: the 7 days of the week and the 24 one-hour periods in a day. A significant amount of bicycles (local test inspired by the computation of a p-value), that are rented at the (departure) station ds on day d (e. g., Monday) at hour h (e. g., from 1pm to 2pm) and returned at the (arrival) station as, translates to an edge from ds to as in the graph timestamped with (d, h). In other terms, the (ds, as, d, h) in the related relation $\mathcal{R}_{\texttt{Vélov'v}} \subseteq \texttt{Departure} \times \texttt{Arrival} \times \texttt{Day} \times \texttt{Hour}$. In the end, this contains $117,411$ 4-tuples, hence a $\frac{117,411}{7\times24\times327\times327} = 0.7\%$ density.

We analyze the results of the experiments with regard to the following questions: (a) Do the discovered graph rules make sense? (b) How to handle time in these rules? (c) What does the exclusive confidence definition capture?, and (d) How does GEAR behave with respect to the parameter settings?

We first searched for rules with time periods and departure stations (tail vertices) at their antecedents; day information at their consequents. In this way, stations that, at some time, "emit" bicycles towards many other stations, but exclusively for some days, are discovered. With the minimum thresholds $\mu = 0.08$ and $\beta = 0.6$, 35 rules are extracted. Fig. 4 reports three of them. The rule in Fig. 4a means that most of the departures from Station 6002 and between 11am and 12am occur on Sundays ($c = 0.71$). This makes sense: this station is at the main entrance of the most popular park, where people like to ride on Sundays. The rule in Fig. 4b means that there rarely are departures from Station 1002 between 1am and 3am except on Sundays ($c = 0.62$). This makes sense: this station is located in a district with many pubs and the favored evenings to party are on Saturdays. Furthermore the public transportation services stop at midnight and the `Vélo'v` is a good alternative to come back home. The rule in 4c describes another known behavior. Many people living outside Lyon arrive by train between 8am and 9am and use `Vélo'v` to finish their trips towards their working place. Indeed, Station 3001 is at the train station inside the main working district. This behavior is specific to the working days ($c = 0.66$).

To answer the question "*which are the stations that often exchange bicycles?*", we searched for rules whose antecedents are departure stations (i. e., tail vertices) and consequents are arrival stations (i. e., head vertices). The support domain of these rules are the Cartesian product of the seven days and the 24 hours. The constraint $\mathcal{C}_{2,2\text{-min-sizes}}$ (see Sect. 4) is additionally enforced, i. e., every rule must involve at least two departure stations and two arrival stations. With $\mu = 0.03$ and $\beta = 0.8$, GEAR returns 27 rules. Fig. 5 reports some of them.

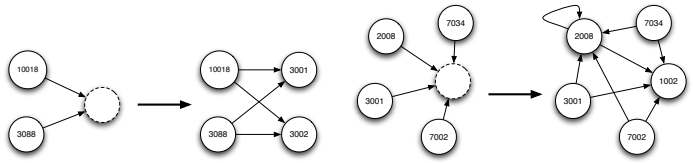


$\{6002\} \times \{11\text{-}12am\} \rightarrow \{Sun.\}$ $(s = 0.09,\ c = 0.71)$ (a)

$\{1002\} \times \{1\text{-}2am, 2\text{-}3am\} \rightarrow \{Sun.\}$ $(s = 0.09,\ c = 0.62)$ (b)

$\{3001\} \times \{8\text{-}9am\} \rightarrow \{working\ days\}$ $(s = 0.13,\ c = 0.66)$ (c)

Fig. 4: Example of rules of the form `Departures` × `Hours` → `Days`.



$\{10018, 3088\} \rightarrow \{3002, 3001\}$ $(s = 0.05,\ c = 0.86)$ (a)

$\{2008, 3001, 7002, 7034\} \rightarrow \{1002, 2008\}$ $(s = 0.04,\ c = 0.82)$ (b)

Fig. 5: Example of rules of the form `Departures` → `Arrivals`.

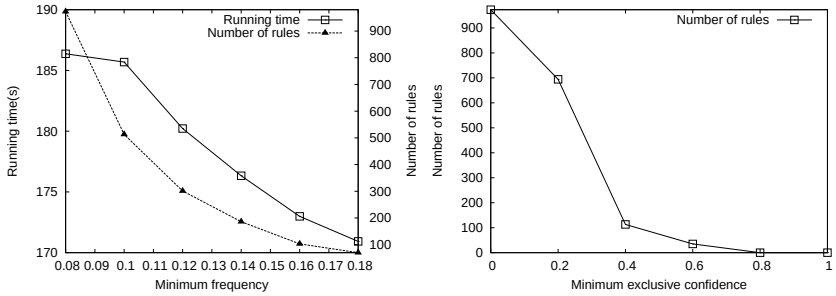
Do some stations exchange many bicycles at favored hours every day? To answer to this question, we search for rules whose antecedents consist of time periods and departure stations (i. e., tail vertices); their consequents are arrival stations (i. e., head vertices). To discover rules that hold every day, the minimal frequency threshold is set to 1. With $\beta = 0.8$, GEAR returns 40 rules which contain at least one time period, two departure stations and two arrival stations. These rules mean that there are some known time periods in which set of stations maintain some privileged bicycle exchanges. Some of them are given in Fig. 6. This kind of knowledge is valuable for the data owner. For instance, if there is no available bicycle at a `Vélo'v` station then other `Vélo'v` stations that maintain strong exchanges with it may be impacted as well.

$\{4\text{-}5pm, 5\text{-}6pm, 7\text{-}8pm\} \times \{1002, 2026\} \rightarrow \{1002, 2026\}$ $(s = 1,\ c = 0.82)$ (a)

$\{5\text{-}6pm\} \times \{3001, 10048\} \rightarrow \{3001, 10048\}$ $(s = 1,\ c = 1)$ (b)

Fig. 6: Example of rules of the form `Hours` × `Departures` → `Arrivals`.

We now report a performance study of GEAR discovering, in $\mathcal{R}_{\texttt{Vélo'v}}$, every frequent inter-dimensional rule of the form $\texttt{Departures} \times \texttt{Hours} \rightarrow \texttt{Days}$. When the minimal frequency threshold increases, the number of frequent associations and the running time decrease (Fig. 7a obtained with $\beta = 0$). Indeed, GEAR prunes large areas of the search space where every association violates the constraint $\mathcal{C}_{\lceil \mu \times |D^{\text{supp}}| \rceil\text{-freq}}$. When the minimum confidence threshold increases, the number of rules decreases too (Fig. 7b obtained with $\mu = 0.08$). GEAR's scalability was tested on the extraction of these rules (still with a frequency exceeding 0.08). To do so the nodes of the graphs were replicated, up to ten times, with their incoming edges only. It turns out that the algorithm scales linearly. More precisely a linear regression of $R \mapsto \frac{T_R}{T_1}$ (where R is how many times the arrival stations are replicated; T_R the running time on this replicated dataset) gives $y = 0.88x + 0.08$ with 0.05 as a standard error. Since $0.88 < 1$, it can be written that GEAR conforms to the proportions of the relation for faster extractions.



(a) Running time w.r.t. μ.

(b) Number of rules w.r.t. β.

Fig. 7: Effectiveness of GEAR.

6 Related Work

Mining graphs has recently received a lot of attention in the data mining community. Many different techniques (e. g., densification laws, shrinking diameter, factorization, clustering, evolution of communities, etc.) [2, 15, 9, 17, 13, 11, 3, 16, 5]. In this section, we focus on methods that mine local patterns. [6] extracts such patterns in labeled dynamic graphs. Frequent subgraph mining algorithms are adapted to time series of graphs. The approach aims at finding subgraphs that are topologically frequent and show an identical dynamic behavior over time, i. e., insertions and deletions of edges occur in the same order of time. Due to the complexity of the task, their algorithm is not complete. Computing the overlap-based support measure means solving a maximal independent set problem and this approach uses a greedy algorithm. [10] proposes a fast algorithm

to mine frequent transformation subsequences from a set of dynamic labeled graphs (the labels on vertices and edges can change over time). Starting with the hypothesis that the changes in a dynamic graph are gradual, they propose to succinctly represent the dynamics with a graph grammar: each change between two observed successive graph states is interpolated by axiomatic transformation rules. [18] studies how a graph is structurally transformed through time. The proposed method computes graph rewriting rules that describe the evolution of two consecutive graphs. These rules are then abstracted into patterns representing the dynamics of a sequence of graphs. [12] introduces the periodic subgraph mining problem, i. e., identifying every frequent closed periodic subgraph. They empirically demonstrate the efficiency and the interest of their proposal on several real-world dynamic social networks. By showing that dynamic graphs can be represented as ternary relations, [8] describes a constraint-based mining approach to discover maximal cliques that are preserved over almost-contiguous timestamps. The constraints are pushed into a closed n-ary pattern mining algorithm. [14] proposes a constraint-based approach too. It the evolution of dense and isolated subgraphs defined by two user-parameterized constraints. Associating a temporal event type with each pattern captures the temporal evolution of the identified subgraph, i. e., the formation, dissolution, growth, diminution and stability of subgraphs between two consecutive timestamps. The algorithm incrementally processes the time series of graphs. [4] introduces the problem of extracting graph evolution rules satisfying minimal support and confidence constraints. It finds isomorphic subgraphs that match the timestamps associated with each edge, and, if present, the properties of the vertices and edges of the dynamic graph. Graph evolution rules are then derived with two different confidence measures. This approach is the closest to ours: it aims at describing a time-evolving graph with rules. Nevertheless, this work focuses on the dynamic changes in the graph whereas we provide a generic framework to discover inter-dimensional rules where the time is either in the rule or in its support.

7 Conclusion

We tackled the problem of describing dynamic graphs via rules that can involve subsets of any dimension (including temporal dimensions) at its antecedent or consequent. We proposed a new semantics for inter-dimensional rules in dynamic graphs. It relies on a relevant objective interestingness measure called the exclusive confidence. We introduced and implemented Gear, an effective solution for computing such rules. Experiments on a real-world dynamic graph demonstrated the interest of our proposal. A timely challenge is to look for primitive constraints that can support more sophisticated knowledge discovery processes in dynamic graphs. Some of these constraints would deal with the temporal dimension(s) (e. g., time contiguity [8]). Other constraints would deal with the "form" of the patterns to discover (e. g., cliques, dense subgraphs, etc.). Another challenge is to revisit, in our setting, important techniques developed for classical association rules, for instance, non redundancy aspects (see, e. g., [19]).

Acknowledgements. This work was partly funded by the ANR project BINGO2 (MDCO 2007) and by a grant from the Vietnamese government.

References

1. Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., Verkamo, A.I.: Fast discovery of association rules. In: Advances in Knowledge Discovery and Data Mining, pp. 307–328. AAAI/MIT Press (1996)
2. Backstrom, L., Huttenlocher, D., Kleinberg, J., Lan, X.: Group formation in large social networks: membership, growth, and evolution. In: KDD. pp. 44–54. ACM Press (2006)
3. Berger-Wolf, T.Y., Saia, J.: A framework for analysis of dynamic social networks. In: KDD. pp. 523–528. ACM Press (2006)
4. Berlingerio, M., Bonchi, F., Bringmann, B., Gionis, A.: Mining graph evolution rules. In: ECML/PKDD. pp. 115–130. Springer (2009)
5. Berlingerio, M., Coscia, M., Giannotti, F., Monreale, A., Pedreschi, D.: As time goes by: Discovering eras in evolving social networks. In: PAKDD (1). pp. 81–90. Springer (2010)
6. Borgwardt, K.M., Kriegel, H.P., Wackersreuther, P.: Pattern mining in frequent dynamic subgraphs. In: ICDM. pp. 818–822. IEEE Computer Society (2006)
7. Cerf, L., Besson, J., Robardet, C., Boulicaut, J.F.: Closed patterns meet n-ary relations. ACM Trans. on Knowledge Discovery from Data 3(1), 1–36 (2009)
8. Cerf, L., Nguyen, T.B.N., Boulicaut, J.F.: Discovering relevant cross-graph cliques in dynamic networks. In: ISMIS. pp. 513–522. Springer (2009)
9. Chi, Y., Zhu, S., Song, X., Tatemura, J., Tseng, B.L.: Structural and temporal analysis of the blogosphere through community factorization. In: KDD. pp. 163–172. ACM Press (2007)
10. Inokuchi, A., Washio, T.: A fast method to mine frequent subsequences from graph sequence data. In: ICDM. pp. 303–312. IEEE Computer Society (2008)
11. Kumar, R., Novak, J., Tomkins, A.: Structure and evolution of online social networks. In: KDD. pp. 611–617. ACM Press (2006)
12. Lahiri, M., Berger-Wolf, T.Y.: Mining periodic behavior in dynamic social networks. In: ICDM. pp. 373–382. IEEE Computer Society (2008)
13. Leskovec, J., Kleinberg, J., Faloutsos, C.: Graphs over time: densification laws, shrinking diameters and possible explanations. In: KDD. pp. 177–187. ACM Press (2005)
14. Robardet, C.: Constraint-based pattern mining in dynamic graphs. In: ICDM. pp. 950–955. IEEE Computer Society (2009)
15. Sun, J., Papadimitriou, S., Yu, P.S., Faloutsos, C.: Graphscope: Parameter-free mining of large time-evolving graphs. In: KDD. pp. 687–696. ACM Press (2007)
16. Tantipathananandh, C., Berger-Wolf, T.Y., Kempe, D.: A framework for community identification in dynamic social networks. In: KDD. pp. 717–726. ACM Press (2007)
17. Tong, H., Papadimitriou, S., Sun, J., Yu, P.S., Faloutsos, C.: Colibri: fast mining of large static and dynamic graphs. In: KDD. pp. 686–694. ACM Press (2008)
18. You, C.H., Holder, L.B., Cook, D.J.: Learning patterns in the dynamics of biological networks. In: KDD. pp. 977–986. ACM Press (2009)
19. Zaki, M.J.: Mining non-redundant association rules. Data Min. Knowl. Discov. 9(3), 223–248 (2004)

Relational Learning of Disjunctive Patterns in Spatial Networks

Corrado Loglisci, Michelangelo Ceci, and Donato Malerba

Department of Computer Science, University of Bari "A.Moro", Italy
{loglisci,ceci,malerba}@di.uniba.it

Abstract. In spatial domains, objects present high heterogeneity and are connected by several relationships to form complex networks. Mining spatial networks can provide information on both the objects and their interactions. In this work we propose a descriptive data mining approach to discover relational disjunctive patterns in spatial networks. Relational disjunctive patterns permit to represent spatial relationships that occur simultaneously with or alternatively to other relationships. Pruning of the search space is based on the anti-monotonicity property of support. The application to the problem of urban accessibility proves the viability of the proposal.

1 Introduction

A spatial network is a network of spatial objects, that is, objects characterized by both a spatial localization (e.g. in a geo-referenced system) and a geometry (e.g. an area). Nodes of spatial networks correspond to spatial objects, while links express spatial relationships (e.g. adjacency). In some cases, links are defined on the basis of other spatial objects (e.g. roads, railways, flights, rivers, etc.). A link might be labeled with a numerical weight which denotes the distance between two nodes. Although spatial networks are of great interest in the study of spatial phenomena, such as urban accessibility, they have not yet received the attention in data mining that they deserve. Yiu and Mamoulis [16] propose the extension of some traditional clustering techniques to face the problem of grouping objects in a large spatial network. In particular, the notion of shortest path between networked nodes is used in partitioning, density-based and hierarchical clustering methods. The same notion of shortest path is exploited in [8] for a problem of outlier detection in a dynamic network, where node insertion/deletion is allowed.

In these, as well as in other related works, a spatial network is modeled as a graph, which simplifies the network by removing the geometry. However, this representation is sometimes oversimplified, since it considers neither the heterogeneity of spatial objects (e.g. public services and private houses should be described by different feature sets) nor the heterogeneity of the spatial relationships expressed in links (e.g. connection by bus, railway or road). This heterogeneity of spatial objects and relationships demands for different representation formalisms and, consequently, a different class of data mining methods which are able to handle this further complexity in the data.

It has been recently argued that the (multi-)relational setting [6] is the most suitable for spatial data mining problems, since it can deal with the heterogeneity of spatial objects, it can distinguish their different role (reference or task-relevant), it can naturally represent a large variety of spatial relationships among objects and it can accommodate different forms of spatial autocorrelation [11]. Several spatial data mining methods have been developed according to the multi-relational setting. They concern descriptive and predictive tasks such as subgroup discovery[9], regression[12] and emerging patterns discovery[2].

In this paper, we extend our previous work on the task of spatial association analysis thorough an inductive logic programming (ILP) approach [1, 10]. Both spatial relationships and properties of spatial objects are represented as predicates, while discovered patterns are defined as conjunctions of atomic formulas built using these predicates. For instance, the following are two examples of spatial patterns which are discovered by the ILP system SPADA [10]:

$\langle district(A),\ road(B),\ intersects(B, A)\rangle$
$\langle district(A),\ road(B),\ crosses(B, A)\rangle$

where the semantics of the two predicates *intersects* and *crosses* is defined by means of the 9-intersection model defined for topological relationships [7]. The support of these patterns is computed by means of a θ-subsumption test[14] against the descriptions of the spatial networks. This is a crisp test which fails when, all other things being equal, two descriptions differ only in the name of a predicate. This brittleness is critical in spatial domains, where the computation of spatial relationships, though supported by a formal semantics, depends on the levels of abstraction granularity. For instance, for slightly different resolutions we may observe either an *intersects* relationship or a *crosses* relationship.

To improve the robustness of the spatial association rule mining method there are two alternatives. First, defining a hierarchy among spatial predicates, which could be used to generalize over spatial relationships. Second, enabling the generation of disjunctive patterns, that is patterns where two or more atoms may be OR-ed to express the variance on the spatial relationship existing between two objects. In this work we follow the second approach, since the definition of a hierarchy among spatial predicates can be cumbersome in many applications. Moreover, in order to prevent the generation of meaningless disjunctions, we exploit a user-defined dissimilarity measure between spatial relationships, which could be used to prune the search space.

The paper is organized as follows. In the next section, related works and contribution of the proposed approach are presented. In Section 3, the approach is presented in detail. In Section 4 the application to a case study is reported. Finally, conclusions are drawn and future works are presented.

2 Related Works and Contribution

Discovering patterns in spatial networks has attracted great interest in the area of in geographical information sciences (GIS). In his seminal work, Zhang [17] introduces a categorization of spatial patterns into grid-like, star-like and irreg-

ular categories of patterns and outlines the differences based on the parallelism relationship among the roads. Less attention has been rather paid in knowledge discovery, although the research in spatial data mining is become mature. A representative work is reported in [4] where the authors propose a framework which extracts snippets from Web, recognizes the locations and, finally, discovers patterns in the form of access points to the recognized locations.

The aforementioned works have common characteristic: spatial objects and networks are represented as vectors or graphs, which suffer from several limitations when heterogeneous spatial objects and relationships have to be represented. This motivates our interest in relational approaches to spatial pattern discovery. Moreover, to cope with the problem of brittleness of subsumption tests for relational patterns, we extend relational mining algorithms in order to discover disjunctive patterns, where alternative relationships between two spatial objects are allowed.

In the literature on frequent pattern mining, we found two noticeable contributions to the problem of discovering disjunctive patterns. In [13], association rules with inclusive or exclusive logical disjunction are discovered, while in [15] traditional algorithms are extended to mine association rules with item groups, where an item group is a disjunction of items created by considering the conceptual distance between items. Both methods work on propositional representations, which are too restrictive for spatial domains.

This paper makes a contribution to the literature on spatial network mining by considering disjunctive patterns in relational formalisms, which are more appropriate to represent spatial networks with heterogeneous spatial objects and relationships. In particular, we extend a method for spatial association rule discovery in order to represent:

1. Disjunctions (e.g. $(intersects(B, A)\ OR\ crosses(B, A))$). They are created by exploiting a user-defined background knowledge in the form of a semantic graph, where vertices correspond to spatial relationships (e.g. *intersects*), while edges denote the semantic relatedness among them and are labelled with numerical weights which quantify the dissimilarity among the relationships (e.g., $intersects \overset{0.9}{\leftrightarrow} crosses$);
2. Disjunctive patterns (e.g., $\langle district(A), transport\_line(B), (intersects(B, A)\ OR\ crosses(B, A)), is\_a(A, market\_square), is\_a(B, road)\rangle$). They are extracted from a graph of patterns which is refined until user-defined input criteria are met.

The proposed approach follows a three-stepped procedure. First, it extracts the infrequent conjunctive patterns which can be upgraded to the disjunctive form. For instance, given $P_1 : \langle district(A), contained\_in(A, B), marketplace(B)\rangle$ and the similarity between *contained_in* and *overlaps*, we can upgrade it to $P_1' : \langle district(A), (contained\_in(A, B)\ OR\ overlaps(A, B)),\ marketplace(B)\rangle$. Second, background knowledge is accommodated to exploit the information on the (dis)similarity among the spatial relationships in the process of generation of disjunctive patterns. Third, disjunctive patterns are produced by iteratively integrating disjunctions into the patterns by means of a pair-wise joining. For instance, given the patterns $P_1 : \langle district(A), contained\_in(A, B), marketplace(B)\rangle$,

$P_2 : \langle district(A), overlaps(A, B), marketplace(B)\rangle$ and assumed that *contained_in* and *overlaps* are two "similar" atoms according to background knowledge, P_1 and P_2 are merged to form the pattern:

$\langle district(A), (contained\_in(A, B)\ OR\ overlaps(A, B)), marketplace(B)\rangle$.

Finally, only the disjunctive patterns whose frequency exceeds the traditional minimum threshold are considered.

3 Learning Disjunctive Relational Patterns

Before formally stating the data mining problem, we introduce some basic notions. In the relational setting, when handling spatial objects, different roles can be played by different *sorts* of data. In a spatial network, objects can be distinguished into target objects of analysis (TO) and non-target objects of analysis (NTO). By introducing this distinction we follow the usual practice in statistics of distinguishing between units of analysis and units of observation. Generalization concerns the units of analysis, while the units of observation are typically secondary data considered potentially useful to explain a phenomenon.

In this work, the target objects (units of analysis) are data on which patterns are enumerated and contribute to compute the frequency of a pattern, while the non-target objects (units of observation) contribute to define the former and can be involved in a pattern. We denote the set of TO as S and the sets of NTO by means of the sets R_k $(1 \leq k \leq M)$, where M is the number of sorts of data that are not considered to be TO. NTOs, belonging to a set R_k, can be organized hierarchically according to a user defined taxonomy. Target objects and non-target objects are represented in Datalog language [3] as ground atoms and populate the extensional part D_E of a deductive database D. A ground atom is an n-ary logic predicate symbol applied to n constants.

Some predicate symbols are introduced in order to express both properties and relationships of TO and NTO. The predicate symbols represent spatial relationships and can be categorized into four classes: 1) *key predicate* identifies the TO in D_E (e.g., in the examples above, *district(·)*); 2) property predicates are binary predicates which define the values taken by an attribute of a TO or of an NTO; 3) structural predicates are binary predicates which relate NTO as well as TO with others NTO (e.g., in the examples above, $contained\_in(\cdot,\cdot)$); 4) *is_a* predicate is a binary taxonomic predicate which associates NTO with a symbol contained in the user defined taxonomy.

The intensional part D_I of the deductive database D includes the definition of the semantic graph (background knowledge) that permits us to express the dissimilarity among spatial relationships in the form of *Datalog* weighted edges of a graph. An example of the Datalog weighted edge is the following:

external_touch_to - (crosses - 0.88)

It states that the dissimilarity between the relationships *external_touch_to(·,·)* and *crosses(·,·)* is 0.88. More generally, it represents an undirected edge e be-

tween two vertices v_i, v_j (e.g., *external touch to*, *crosses*) with weight w_{ij} (e.g., 0.88) and it is denoted as $e(v_i, v_j, w)$. A finite sequence of undirected links $e_1, e_2, \ldots, e_m$ which connects two vertices v_i, v_j is called *path* and denoted as $\rho(n_i, n_j)$. The complete list of such undirected edges represents the background information on the dissimilarity among relationships and allows to join patterns by introducing disjunctions *(externa_touch_to(A,B) OR crosses(A,B))*.

Discovered patterns are conjunctions of Datalog non-ground atoms and disjunctions of non-ground atoms, which can be expressed by means of a set notation. A Datalog non-ground atom is an n-ary predicate symbol applied to n terms (either constants or variables), at least one of which is a variable. A formal definition of pattern of our interest is reported in the following:

Definition 1. *A* disjunctive pattern P *is a set of atoms and disjunctions of atoms* $p_0(t_0^1)$, $(p_1(t_1^1, t_1^2)|p_2(t_2^1, t_2^2)|\ldots), \ldots, (p_k(t_k^1, t_k^2)|\ldots|p_{k+h}(t_{k+h}^1, t_{k+h}^2))$ *where* p_0 *is the key predicate, while* p_i, $i = 1, \ldots, k+h$, *is either a structural predicate or a property predicate or an* is_a *predicate. Symbol " | " indicates disjunctions.*

Terms t_i^j are either constants, which correspond to values of property predicates, or variables, which identify target objects or non-target objects. Each p_i is a predicate occurring in D_E (extensionally defined predicate). Some examples of disjunctive patterns are the following:
$P_1 \equiv district(A), (comes\_from(A, B)|external\_ends\_at(A, B)), shape(A, rectangle)$
$P_2 \equiv district(A), (external\_ends\_at(A, B)|runs\_along\_boundary\_and\_goes\_in(A, B)),$
$transport\_net(A, roads)$
where the variables A denote target objects, and variables B denote some non-target objects, while the predicates $district(A)$ identify the key predicate in P_1 and P_2, $shape(A, rectangle)$ and $transport\_net(A, roads)$ are property predicates and the others are structural predicates. All variables are implicitly existentially quantified.

We now can give a formal statement of the problem of discovering relational frequent patterns with disjunctions:
1. *Given:* the extensional part D_E of a deductive database D, and two thresholds $minSup \in [0; 1]$, $nSup \in [0; 1]$ ($minSup$ represents a minimum frequency value while $nSup$ represents a maximum frequency value, $nSup < minSup$), *Find:* the collection I_R of the relational infrequent patterns whose support is included in $[nSup; minSup)$.
2. *Given:* the collection I_R, the intensional part D_I of a deductive database D, and two thresholds $minSup$ and $\gamma \in [0; 1]$ (γ defines the maximum dissimilarity value of relationships involved in the disjunctions), *Find:* relational disjunctive patterns whose frequency exceeds $minSup$ and whose dissimilarity of relationships involved in the disjunctions does not exceed γ.

3.1 Discovering Infrequent Conjunctive Patterns

The intuition underlying the discovery of pattern with disjunctions is that of extending infrequent conjunctive patterns with disjunctive forms until the thresh-

old *minSup* is exceeded. Each conjunctive pattern P is associated with a statistical parameter $sup(P, D)$ (support of P on D), which is the percentage of *units of analysis* in D *covered by* P. More precisely, a unit of analysis of a target object $s \in S$ is a subset of ground atoms in D_E defined as follows:

$$D[s] = is\_a(R(s)) \cup D[s|R(s)] \cup \bigcup_{r_i \in R(s)} D[r_i|R(s)], \quad (1)$$

where $R(s)$ is the set of NTO directly or indirectly related to s, $is\_a(R(s))$ is the set of is_a atoms which define the sorts of $r_i \in R(s)$, $D[s|R(s)]$ contains properties of s and relations between s and some $r_i \in R(s)$, $D[r_i|R(s)]$ contains properties of r_i and relations between r_i and some $r_j \in R(s)$. By assigning a pattern P with an existentially quantified conjunctive formula $eqc(P)$ obtained by transforming P into a Datalog query, the units of analysis $D[s]$ are *covered by* a pattern P if $D[s] \models eqc(P)$, namely $D[s]$ logically entails $eqc(P)$).

Conjunctive patterns are mined with SPADA[10] which however enables the discovery of relational patterns whose support exceeds *minSup* (frequent patterns). In this work we exploit the capabilities of SPADA to identify infrequent conjunctive patterns, but this does not exclude the possibility of using other methods for mining infrequent relational patterns in this initial processing step. SPADA performs a breadth-first search of the space of patterns, from the most general to the more specific ones, and prunes portions of the space which contain only infrequent patterns, which are the conjunctive patterns of our interest. The pruning strategy guarantees that all infrequent patterns are removed and, at this aim, uses a generality ordering based on the notion of θ-subsumption [14]:

Definition 2. *P_1 is more general than P_2 under θ-subsumption ($P_1 \succeq_\theta P_2$) if and only if P_1 θ-subsumes P_2, i.e. a substitution θ exists, such that $P_1\theta \subseteq P_2$.*

For instance, given $P1 \equiv district(A), crosses(A, B)$, $P2 \equiv district(A), crosses(A, B), is\_a(B, transport\_net)$, $P3 \equiv district(A), crosses(A, B), is\_a(B, transport\_net), along(A, C)$ we observe that P_1 θ-subsumes P_2 ($P_1 \succeq_\theta P_2$) and P_2 θ-subsumes P_3 ($P_2 \succeq_\theta P_3$) with substitutions $\theta_1 = \theta_2 = \oslash$. The generality order is monotonic with respect to the pattern support, so whenever $P1$ will be infrequent the patterns more specific of it (e.g., $P2, P3$) will be infrequent too.

The search is based on the level-wise method and implements a two-stepped procedure: i) generation of candidate patterns with k atoms (k-th level) by considering the frequent patterns with $k-1$ atoms ((*k-1*)-th level); ii) evaluation of the frequency with k atoms. So, the patterns whose support does not exceeds *minSup* will be not considered for the next level: the patterns discarded (infrequent) at each level are rather considered for the generation of disjunctions. The collection I_R is thus composed of a subset of infrequent patterns, more precisely those with support greater than or equal to *nSup* (and less than *minSup*).

3.2 Upgrading Relational Patterns with Disjunctions

The generation of disjunctive patterns is performed by creating disjunctions among similar relationships (thus similar atoms in the patterns) in accordance

to the background semantic graph: two patterns which present similar atoms are joined to form only one. The implemented algorithm (see Algorithm 1) is composed of two sub-procedures: the first one (lines 2-12) creates a graph $\mathcal{G}_\mathcal{D}$ with the patterns of I_R by exploiting the knowledge defined in D_I, while the second one (lines 13-32) joins two patterns (vertices) on the basis of the information (weight) associated to their edge. The initial graph $\mathcal{G}_\mathcal{D}$ evolves due to joining of patterns on the vertices until the setting of $minSup$ and γ is met (Section 3.1).

In particular, for each pair of patterns which have the same length (namely, at the same level of the level-wise search method) it checks whether they differ in only one atom and share the remaining atoms up to a redenomination of variables (line 3). Let α and β be the two atoms differentiating P from Q (α in P, β in Q), a path ρ which connects α to β (or viceversa) is searched among the weighted edges according to D_I (semantic network): in the case the sum ω of the weights found in the path is lower than the maximum dissimilarity γ the vertices P and Q are inserted into $\mathcal{G}_\mathcal{D}$ and linked through an edge with weight ω (lines 4-9). Note that when there is more than one path between α and β, then the path with lowest weight is considered. Intuitively, at the end of the first sub-procedure, $\mathcal{G}_\mathcal{D}$ will contain, as vertices, the patterns which meet the condition at the line 3, and it will contain, as edges, the weights associated to the path linking the atoms differentiating the patterns.

Once we have $\mathcal{G}_\mathcal{D}$, a list $\mathcal{L}_\mathcal{D}$ is populated with the vertices and edges of $\mathcal{G}_\mathcal{D}$: an element of $\mathcal{L}_\mathcal{D}$ is a triple $\langle P, Q, \omega \rangle$ composed of a pair of vertices-patterns (P,Q) with their relative weight. Elements in $\mathcal{L}_\mathcal{D}$ are ranked in ascending order with respect to the values of ω so that the pairs of patterns with lower dissimilarity will be joined for first. This guarantees that disjunctions with very similar atoms will be preferred to the others (line 13). For each element of $\mathcal{L}_\mathcal{D}$ whose weight ω is lower than γ the two patterns P, Q are joined to generate a pattern J composed by the conjunction of the same atoms in common to the two patterns P, Q and of the disjunction formed by the two different (but similar) atoms (lines 14-15). This joining procedure permits to have patterns with the same length of the original ones and which occur when at least one of original patterns occurs. Therefore, if a pattern J is obtained by joining P and Q, it covers a set of units of analysis equal to the union of those of P and Q: the support of J is determined as in line 16 and, generally, it is higher than the support of P and Q. In the case the support of J exceeds $minSup$ then it can be considered statistically interesting and no further processing is necessary (lines 16-17). Otherwise, J is again considered and inserted into $\mathcal{G}_\mathcal{D}$ as follows. The edges which linked another pattern R of $\mathcal{G}_\mathcal{D}$ to P and Q are modified in order to keep the links from R to J: the weight of the edges between one pattern R and J will be set to the average value of the weights of all the edges which linked R to P and Q (lines 19-27). The modified graph $\mathcal{G}_\mathcal{D}$ contains conjunctive patterns (those of I_R) and pattern with disjunctions (those produced by joining). Thus, $\mathcal{G}_\mathcal{D}$ is re-evaluated for further joins and the algorithm proceeds iteratively (line 29-30) until no additional disjunctions can be done (namely, when $\mathcal{L}_\mathcal{D}$ is empty or the weights ω are higher than γ). At each iteration, the patterns P and Q are removed from $\mathcal{G}_\mathcal{D}$ (line 32).

Algorithm 1 Upgrading Relational Pattern with Disjunctions.

input: $I_R, D_I, \gamma, minSup$ **output:** $\mathcal{J}$ // $\mathcal{J}$ set of disjunctive patterns
for all $(P,Q) \in I_R \times I_R$, $Q \neq P$ **do**
  if $P.length = Q.length$ and $check\_atoms(P,Q)$ **then**
    $(\alpha, \beta) := atoms\_diff(P,Q)$ //α, β atoms differentiating P,Q
    if $\rho(\alpha,\beta) \neq \emptyset$ **then**
      $\omega := \sum_{e(v_i,v_j,w_{ij}) \ in \ \rho(\alpha,\beta)} w_{ij}$
      if $\omega \leq \gamma$ **then**
        $addNode(P, \mathcal{G}_\mathcal{D})$; $addNode(Q, \mathcal{G}_\mathcal{D})$; $addEdge(P, Q, \omega, \mathcal{G}_\mathcal{D})$
      end if
    end if
  end if
end for
$\mathcal{L}_\mathcal{D} \leftarrow$ edges of $\mathcal{G}_\mathcal{D}$ // list of edges of $\mathcal{G}_\mathcal{D}$ ordered in ascending mode w.r.t. ω
while $\mathcal{L}_\mathcal{D} \neq \emptyset$ and $\forall e(P,Q,\omega) \in \mathcal{G}_\mathcal{D}$ $\omega \leq \gamma$ **do**
  $J \leftarrow join(P,Q)$; $J.support := P.support + Q.support - (P \cap Q).support$;
  if $J.support \geq minSup$ **then**
    $\mathcal{J} := \mathcal{J} \cup \{J\}$
  else
    for all R *such that* $\exists\ e(P,R,\omega_1) \in \mathcal{G}_\mathcal{D}$ and $\exists\ e(Q,R,\omega_2) \in \mathcal{G}_\mathcal{D}$ **do**
      $addEdge(R, J, (\omega_1 + \omega_2)/2, \mathcal{G}_\mathcal{D})$
    end for
    for all R *such that* $\exists\ e(P,R,\omega_1) \in \mathcal{G}_\mathcal{D}$ and $\nexists\ e(Q,R,\omega_2) \in \mathcal{G}_\mathcal{D}$ **do**
      $addEdge(R, J, \omega_1, \mathcal{G}_\mathcal{D})$
    end for
    for all R *such that* $\exists\ e(Q,R,\omega_2) \in \mathcal{G}_\mathcal{D}$ and $\nexists\ e(P,R,\omega_1) \in \mathcal{G}_\mathcal{D}$ **do**
      $addEdge(R, J, \omega_2, \mathcal{G}_\mathcal{D})$
    end for
    $\mathcal{L}_\mathcal{D} \leftarrow$ edges of $\mathcal{G}_\mathcal{D}$
    update $\mathcal{L}_\mathcal{D}$
  end if
  $removeNode(P, \mathcal{G}_\mathcal{D})$; $removeNode(Q, \mathcal{G}_\mathcal{D})$
end while

An explanatory example is illustrated in Figure 1. Consider the background knowledge D_I on the dissimilarity among four spatial relationships and the set I_R containing four infrequent conjunctive patterns as illustrated in Figure 1a and γ equal to 0.7. The first sub-procedure of the algorithm 1 analyzes P_1, P_2, P_3, P_4 and discovers that they differ in only one atom, while the other atoms are in common. Then, it creates the graph $\mathcal{G}_\mathcal{D}$ by collocating P_1, P_2, P_3 in three different vertices and linking them through edges whose weights are taken from the paths ρ in D_I. P_4 is not considered because the vertex *overlaps* has dissimilarity with *internal_ends_at* higher than γ (row (1) in Figure 1b). The second sub-procedure starts by ordering the weights of the edges: the first disjunction is created by joining P_1 and P_3 given that the dissimilarity value is lower than γ and the lowest (row (2) in Figure 1b). Next, the pattern so created

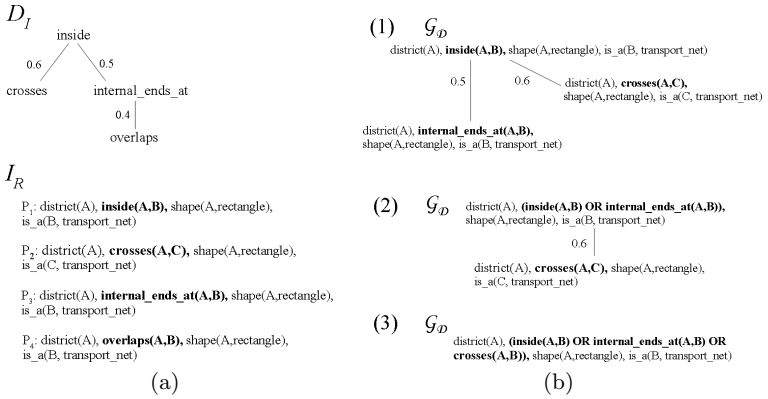


Fig. 1. Extending relational pattern with disjunctions: an example (γ=0.7).

and P_2 are checked for joining. Both have the same length and differ in only one atom. Although the first presents a disjunction and the second presents a "simple" atom, dissimilarity is lower than γ and a new disjunctive pattern is created (row (3) in Figure 1b).

4 Experiments

This approach has been implemented as the extension of the system SPADA aimed to discover relational patterns with disjunctions: the system (afterwards *jSPADA*) is now able to mine conjunctive patterns and disjunctive patterns as well. Here we present the application of both systems to mine spatial networks in a case study of urban accessibility. More precisely, the spatial network is obtained by analyzing both census and digital maps of Stockport, one of the ten districts in Greater Manchester and the analysis is aiming at investigating the accessibility *to* the Stepping Hill Hospital *from* the actual residence of people living far from the hospital. In this case study, transport network, namely the layers of roads, railways and bus priority lines, correspond to the links of the spatial network, while districts close to the hospital and districts distant from the hospital corresponds to the nodes of the network. In accordance with our setting defined in Section 3, districts close to the hospital are target objects while the transport network and districts distant from the hospital are non-target objects.

Property predicates define people with own cars and are *no_car()*, *one_car()*, *two_cars()*, *three_more_cars()*. Structural predicates represent binary topological relationships between districts and roads, railways or bus lines, and correspond to the twelve feasible relations between a region and a line according to the 9-intersection model [7]. Here, background knowledge D_I has been defined on the structural predicates and the dissimilarity values have been manually determined

by applying the Sokal-Michener dissimilarity measure on the matrix representation of the twelve relations[5]: for instance, the following $external\_ends\_at \overset{0.22}{\leftrightarrow} along$; $along \overset{0.277}{\leftrightarrow} comes\_from$ expresses the similarity among three spatial relationships quantified with 0.22 and 0.2777 respectively. Districts and transport network can be involved in more than one line-region spatial relationships and this advocates the usage of disjunctive patterns. D_E contains 1147 ground atoms for 152 target objects.

Experiments were performed[1] by tuning the thresholds $minSup$, $nSup$, γ and the results are reported in Figure 2. A comparison between SPADA and jSPADA has been conducted by varying $minSup$, while, for jSPADA, the values of $nSup$ and γ are set to 0.005 and 0.6 respectively. As we can see the histogram values reported in Figure 2a, jSPADA discovers a number of patterns that is higher than that of SPADA. Indeed, jSPADA returns a set which includes those frequent conjunctive (generated by SPADA) and those disjunctive generated by re-evaluating the infrequent conjunctive ones. Thus, as $minSup$ increases, the range $[nSup; minSup)$ becomes larger and, generally, more disjunctive patterns are extracted while the number of conjunctive frequent patterns decreases. It is worthy that the set of only disjunctive patterns (the complement of the set of patterns of jSPADA relative to the set of SPADA) is actually much smaller than the set of only conjunctive patterns (patterns of SPADA). For instance, when $minSup$=0.007 the number of disjunctive patterns amounts to 5, while the number of conjunctive patterns is 898. Thus, the problem of huge amounts of disjunctive patterns is not so relevant as in the case of conjunctive patterns. This is a clear advantage of the proposed approach since the classical problem of manual analysis of patterns is mitigated.

As expected, also the threshold $nSup$ has influence on the patterns discovered by jSPADA. Indeed, from the figures 2c and 2d ($minSup = 0.025$ and $\gamma = 0.6$) we note that jSPADA is highly sensitive to $nSup$ since the number of disjunctive patterns is reduced of one order of magnitude (from 20 to 0) while $nSup$ is increased by factor of two (from 0.01 to 0.02). By comparing the plots (a), (c) and (d) we note that, by varying $minSup$, jSPADA has a limited capacity in unearthing infrequent patterns (but potentially interesting) than when varying $nSup$. This confirms the viability of the approach to discover new forms of interesting patterns. The sensitivity of the algorithm can be evaluated with respect to the dissimilarity of the disjunctions (Figure 2b). At high values of γ disjunctions can be created also between relationships whose similarity is small, so the patterns present disjunctions with several atoms and the final set is larger. On the contrary, lower values of γ permit to identify disjunctions only between very similar relationships, so the disjunctions present less atoms and the final set is smaller: when γ is set to 0.4, no disjunction is created since the minimum value of similarity between relationships amounted to 0.44.

A comparison between jSPADA and SPADA can also be done from a qualitative viewpoint. jSPADA enables the discovery of patterns which enrich the information extracted by SPADA. For instance, the pattern discovered by SPADA

[1] Data and results are accessible at http://www.di.uniba.it/∼loglisci/jSPADA/

$P_1 : district(A), comes\_from(A, B), is\_a(B, road), comes\_from(A, C), is\_a(C, road)$ [$support$: 12%]

is enriched by P_2 discovered by jSPADA:

$P_2 : district(A), [comes\_from(A, C)|external\_ends\_at(A, C)], is\_a(C, road),$
$comes\_from(A, B), is\_a(B, rail)$ [$support$: 16%]

which introduces the disjunctions $comes\_from(A, C)|external\_ends\_at(A, C)$ between two structural predicates. P_2 expresses the information that the road named as C can be connected to the district named as A through two possible simultaneous or alternative ways, $comes\_from(A, C)$ (C starts in A and terminates outside A) and $external\_ends\_at(A, C)$ (C starts outside A and terminates inside A). Remarkably, the support of P_2 is higher than that of P_1. jSPADA permits also the discovery of completely novel patterns that SPADA neglects. One of these is the following:

$P_3 : district(A), [external\_ends\_at(A, B)|along(A, B)|comes\_from(A, B)],$
$three\_more\_cars(A, [0.033; 0.114])$ [$support$: 11.1%]

which introduces a property predicate (i.e., the percentage of households owing more three cars included in [0.033;0.114]) and expresses in the disjunction three possible forms of accessibility to the district A by the transport line B.

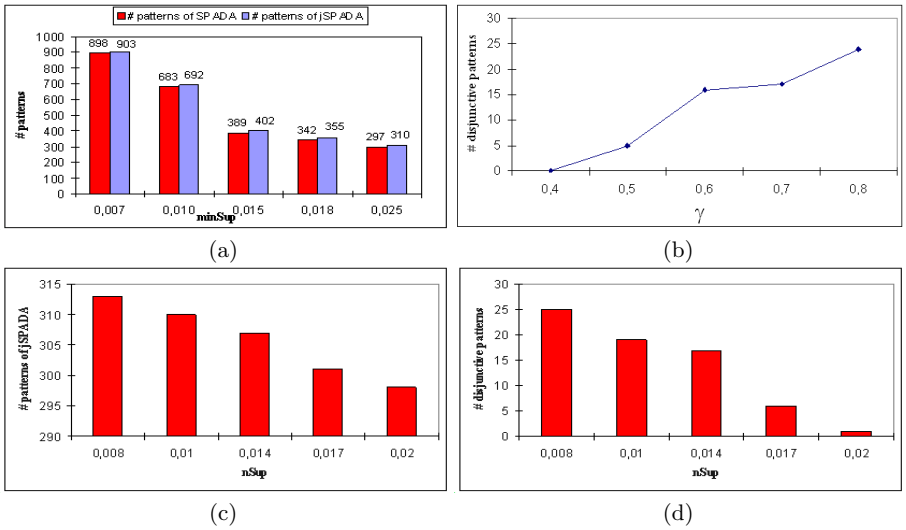


Fig. 2. Number of patterns discovered by tuning $minSup$, $nSup$, γ.

5 Conclusion

In this paper we present a relational data mining approach to discover disjunctive frequent patterns in spatial networks when considering a variance of spatial

relationships existing between two objects. The introduction of disjunctions into the patterns permits to represent spatial relationships which occur simultaneously with or alternatively to others. The application to the problem of urban accessibility points out some peculiarities of the proposal. As future work, we intend to extend experiments to evaluate scalability of the approach.

Acknowledgment. This work is supported by the Strategic Project PS121 "Telecommunication Facilities and Wireless Sensor Networks in Emergency Management" funded by Apulia Region.

References

1. A. Appice, M. Ceci, A. Lanza, F. A. Lisi, and D. Malerba. Discovery of spatial association rules in geo-referenced census data: A relational mining approach. *IDA*, 7(6):541–566, 2003.
2. M. Ceci, A. Appice, and D. Malerba. Discovering emerging patterns in spatial databases: A multi-relational approach. In *PKDD*, volume 4702 of *LNCS*, pages 390–397, 2007.
3. S. Ceri, G. Gottlob, and L. Tanca. *Logic Programming and Databases*. Springer, 1990.
4. D. Davidov and A. Rappoport. Geo-mining: discovery of road and transport networks using directional patterns. In *EMNLP '09*, pages 267–275, 2009.
5. E. Diday and F. Esposito. An introduction to symbolic data analysis and the sodas software. *Intell. Data Anal.*, 7(6):583–601, 2003.
6. S. Dzeroski and N. Lavrac. *Relational Data Mining*. Springer-Verlag, 2001.
7. M. J. Egenhofer and R. D. Franzosa. Point set topological relations. *International Journal of Geographical Information Systems*, 5:161–174, 1991.
8. W. Jin, Y. Jiang, W. Qian, and A. K. H. Tung. Mining outliers in spatial networks. In *DASFAA*, volume 3882 of *LNCS*, pages 156–170. Springer, 2006.
9. W. Klösgen and M. May. Spatial subgroup mining integrated in an object-relational spatial database. In *PKDD*, volume 2431 of *LNCS*, pages 275–286, 2002.
10. F. A. Lisi and D. Malerba. Inducing multi-level association rules from multiple relations. *Machine Learning*, 55(2):175–210, 2004.
11. D. Malerba. A relational perspective on spatial data mining. *IJDMMM*, 1(1):103–118, 2008.
12. D. Malerba, M. Ceci, and A. Appice. Mining model trees from spatial data. In *PKDD*, volume 3721 of *LNCS*, pages 169–180, 2005.
13. A. A. Nanavati, K. P. Chitrapura, S. Joshi, and R. Krishnapuram. Mining generalised disjunctive association rules. In *CIKM*, pages 482–489. ACM, 2001.
14. G. D. Plotkin. A note on inductive generalization. *Mach. Intell.*, 5:153–163, 1970.
15. J. F. Roddick and P. Fule. Semgram - integrating semantic graphs into association rule mining. In *AusDM*, volume 70 of *CRPIT*, pages 129–137, 2007.
16. M. L. Yiu and N. Mamoulis. Clustering objects on a spatial network. In *SIGMOD Conference*, pages 443–454. ACM, 2004.
17. Q. Zhang. Modeling structure and patterns in road network generalization. *7th ICA Workshop on Generalization*, 2004.

Spectral Co-Clustering for Dynamic Bipartite Graphs

Derek Greene, Pádraig Cunningham

School of Computer Science & Informatics, University College Dublin
{derek.greene,padraig.cunningham}@ucd.ie

Abstract. A common task in many domains with a temporal aspect involves identifying and tracking clusters over time. Often dynamic data will have a feature-based representation. In some cases, a direct mapping will exist for both objects and features over time. But in many scenarios, smaller subsets of objects or features alone will persist across successive time periods. To address this issue, we propose a dynamic spectral co-clustering method for simultaneously clustering objects and features over time, as represented by successive bipartite graphs. We evaluate the method on a benchmark text corpus and Web 2.0 bookmarking data.

1 Introduction

In many domains, where the data has a temporal aspect, it will be useful to analyse the formation and evolution of patterns in the data over time. For instance, researchers may be interested in tracking evolving communities of social network users, such as clusters of frequently interacting authors in the blogosphere, or circles of users with shared interests on social media sites. In the case of online news sources, producing large volumes of articles on a daily basis, it will often be useful to chart the development of individual news stories over time.

For many of these problems it may be of interest to simultaneously identify clusters of both data objects and features. This task, often referred to as *co-clustering*, has been formulated as the problem of partitioning a bipartite graph, where the two types of nodes correspond to objects and features [1]. However, to the best of our knowledge, this work has been limited to static applications, where temporal information is unavailable or has been disregarded.

A popular recent approach to the problem of clustering dynamic data has been to use an "offline" strategy, where the dynamic data is divided into discrete *time steps*. Sets of *step clusters* are identified on the individual time steps using a suitable clustering algorithm, and these step clusters are associated with one another over successive time steps [2]. However, clusters may change considerably between time steps. This can be problematic, both for the purpose of matching clusters between time steps, and for supporting users to follow and understand how groups are changing over time. To address this problem, both current and historic information can be incorporated into the objective of the

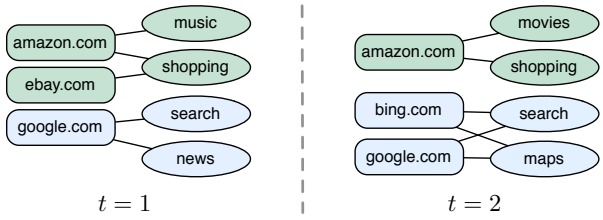


Fig. 1. A dynamic co-clustering scenario where 2 clusters appear in 2 time steps. Note that a subset of both objects (bookmarks) and features (tags) persists across time.

clustering process [3]. Benefits of this approach include increasing the smoothness of transitions between clusterings over time, and improving cluster quality by incorporating historic information to reduce the effects of noisy data.

A number of additional considerations arise when tracking dynamic data represented in feature spaces. Notably, a set of objects or features will not always persist in the data across steps. In general, three different scenarios are possible:

1. Data objects alone persist across time steps. For instance, in bibliographic networks, papers are only published at a single point in time, whereas authors will generally be present in the network over an extended period of time.
2. Features alone persist across time. In a news collection, articles will appear once, whereas terms may continue to appear as topics extend over time.
3. Both objects and features persist across time. For example, in the case of Web 2.0 tagging portals, both the individual tags and the objects being tagged (*e.g.* bookmarks, images) will appear in multiple time steps. A simple example with just two clusters is shown in Figure 1.

Here we consider the problem of tracking nodes in multiple related dynamic bipartite graphs. In Section 3 we describe the main contribution of this paper – a dynamic spectral co-clustering algorithm for simultaneously grouping objects and features over time, in any of the above scenarios. This algorithm takes into account both information from the current time step, together with historic information from the previous step. In our evaluations in Section 4 we show that the proposed algorithm works both in the case where features alone persist over time, and when objects **and** features persist. These evaluations are performed on a labelled benchmark news corpus and Web 2.0 tagging data.

2 Related Work

2.1 Co-clustering

In certain problems it may be useful to perform *co-clustering*, where both objects and features are assigned to groups simultaneously. One approach to the co-clustering problem is to view it as the task of partitioning a weighted bipartite graph. Dhillon [1] proposed a spectral approach to approximate the optimal

normalised cut of a bipartite graph, which was applied for document clustering. This involved computing a truncated singular value decomposition (SVD) of a suitably normalised term-document matrix, constructing an embedding of both terms and documents, and applying k-means to this embedding to produce a simultaneous k-way partitioning of both documents and terms. Mirzal & Furukawa [4] provided a further theoretical grounding for spectral co-clustering, demonstrating that simultaneous row and column clustering is equivalent to solving the separate row and column clustering problems.

2.2 Dynamic Clustering

The general problem of identifying clusters in dynamic data has been studied by a number of authors. Early work on the unsupervised analysis of temporal data focused on the problems of topic tracking and event detection in document collections [5]. More recently, Chakrabarti *et al.* [3] proposed a general framework for "evolutionary clustering", where both current and historic information was incorporated into the objective of the clustering process. The authors used this to formulate dynamic variants of common agglomerative and partitional clustering algorithms. In the latter case, related clusters were tracked over time by matching similar centroids across time steps. Two evolutionary versions of spectral partitioning for classical (unipartite) graphs were proposed by Chi *et al.* [6]. The first version (PCQ) involved applying spectral clustering to produce a partition that also accurately clusters historic data. The second version (PCM) involved measuring historic quality based on the chi-square distance between current and previous partition memberships.

The application of dynamic clustering methods has been particularly prevalent in the realm of social network analysis, where the goal is to identify communities of users in dynamic networks. Palla *et al.* [7] proposed an extension of the popular CFinder algorithm to identify community-centric evolution events in dynamic graphs, based on an offline strategy. This extension involved applying community detection to composite graphs constructed from pairs of consecutive time step graphs. Another life-cycle model was proposed in [2], where the dynamic community finding approach was formulated as a graph colouring problem. The authors proposed a heuristic solution to this problem, by greedily matching pairs of node sets between time steps. The problem of clustering data over time has also been considered in the temporal analysis domain. Kalnis *et al.* [8] described a density-based clustering approach where clusters persist over time, despite continuous changes in cluster memberships. This corresponds closely to the "assembly line" dynamic clustering scenario described in [2].

3 Methods

3.1 Problem Definition

We represent a dynamic feature-based dataset as a set of l bipartite graphs $\{G_1, \ldots, G_l\}$. Each *step graph* G_t consists of two sets of nodes, representing the

n_t data objects, and m_t features present in the data at time t. Edges exist only between nodes of different types, corresponding to non-zero feature values. We can conveniently represent each step graph using a feature-object matrix $\mathbf{A}_t$ of size $m_t \times n_t$.

In the offline formulation of the dynamic co-clustering problem, the overall goal is to identify a set of *dynamic clusters* of objects and features, which appear in the data across one or more time steps. We refer to *step clusters* that are identified on individual step graphs, which represent specific observations of dynamic clusters at a given point in time. The formulation therefore has two key requirements: a suitable clustering algorithm to cluster individual time step graphs (ideally in a way that incorporates historic information), and an approach to track these clusters across time steps. While our primary focus here is on the former aspect, in Section 3.3 we also briefly discuss the latter aspect.

3.2 Dynamic Spectral Co-clustering

We now introduce a dynamic co-clustering algorithm that considers both historic information from the previous time step, and the internal quality of the clustering in the current time step. The algorithm consists of three phases: bipartite spectral embedding, cluster initialisation, and a cluster assignment phase.

Spectral embedding. Following normalised cut optimisation via spectral co-clustering described in [1], for a given time step feature-object matrix $\mathbf{A}_t$, we construct the degree-normalised matrix $\hat{\mathbf{A}}_t = {\mathbf{D_1}}^{-\frac{1}{2}} \mathbf{A}_t {\mathbf{D_2}}^{-\frac{1}{2}}$, where $\mathbf{D_1}$ and $\mathbf{D_2}$ are diagonal column and row degree matrices respectively. We then apply SVD to $\hat{\mathbf{A}}_t$, computing the leading left and right singular vectors corresponding to the largest singular values. Following the choice made by many authors in the spectral clustering literature, we use k_t dimensions corresponding to the expected number of clusters. Although the issue of selecting the number of clusters is not discussed in this paper, one potential approach is to choose k_t based on the eigengap method [9]. The truncated SVD yields matrices $\mathbf{U_{k_t}}$ and $\mathbf{V_{k_t}}$. A unified embedding of size $(m_t + n_t) \times k_t$ is constructed by normalising and stacking the truncated factors as follows:

$$\mathbf{Z}_t = \begin{bmatrix} {\mathbf{D_1}}^{-1/2} \mathbf{U_{k_t}} \\ {\mathbf{D_2}}^{-1/2} \mathbf{V_{k_t}} \end{bmatrix} \tag{1}$$

Prior to clustering, the rows of $\mathbf{Z}_t$ are subsequently re-normalised to have unit length, as proposed for spectral partitioning in [9]. This process provides us with a k_t-dimensional embedding of all nodes of both types in G_t.

Cluster initialisation. At time $t = 1$, we have no historic information. Therefore to seed the clustering process, we use a variant of orthogonal initialisation as proposed by Ng *et al.* [9] for spectral graph partitioning. This operates using a "farthest-first" strategy as follows. The first cluster centroid is chosen to be the

mean vector of the rows in $\mathbf{Z}_t$. We then repeatedly select the next centroid to be the row in $\mathbf{Z}_t$ that is closest to being 90° from those that have been previously selected. This process continues until k_t centroids have been chosen.

For each time step $t > 1$, we initialise using clusters from the previous time step. A simple approach is to map the clusters generated on the embedding for time $t-1$ to $\mathbf{Z}_t$. However, as noted previously, not all features and objects will persist between time steps. To produce an initial clustering at time t, we identify the intersection of the sets of nodes present in the graphs G_{t-1} and G_t. The clusters containing these are mapped to the embedding $\mathbf{Z}_t$, and we compute the resulting centroids. If less than k_t centroids are produced, the remaining centroids are chosen from the rows of $\mathbf{Z}_t$ using orthogonal selection as above. We can then predict memberships for each unassigned row z_i of $\mathbf{Z}_t$, using a simple nearest centroid classifier to maximise the similarity:

$$\max_{C \in C_t} \; z_i^{\intercal} \mu_c \tag{2}$$

where μ_c is the centroid of cluster C_c. This classification procedure yields a predicted clustering for rows in $\mathbf{Z}_t$ (*i.e.* a co-clustering of all objects and features present at time t), which we denote $\mathcal{P}_t$.

Cluster assignment. To recover a clustering from $\mathbf{Z}_t$, we apply a constrained version of k-means clustering to the rows of the embedding, which takes into account both the internal quality of the current partition, and agreement with the predicted partition $\mathcal{P}_t$. We distinguish the latter from the membership preservation objective described in [6] – here we use predicted memberships for missing objects and features missing from the previous step.

As a measure of current cluster quality, we use vector-centroid similarities as in Eqn. 2. Historical quality is calculated based on the quantity $\text{pred}(\mathcal{P}_t, \mathcal{C}_t)$, which denotes the degree to which the predicted cluster assignments in $\mathcal{P}_t$ agree with those in the current clustering $\mathcal{C}_t$. To quantify this agreement, we use a variant of pairwise *prediction strength* [10]:

$$\text{pred}(\mathcal{P}_t, \mathcal{C}_t) = \sum_{C \in \mathcal{C}_t} \frac{1}{|C|\,(|C|-1)} \sum_{(z_i, z_j) \in C} co(z_i, z_j) \tag{3}$$

where $co(z_i, z_j) = 1$ if both rows were predicted to be coassigned in $\mathcal{P}_t$, or $c(z_i, z_j) = 0$ otherwise.

To combine both sources of information, the clustering objective then becomes a weighted combination of two objectives:

$$J(\mathcal{C}_t) = (1-\alpha) \cdot \left(\sum_{c=1}^{k} \sum_{z_i \in C_c} z_i^{\intercal} \mu_c \right) + \alpha \cdot (\text{pred}(\mathcal{P}_t, \mathcal{C}_t)) \tag{4}$$

This type of aggregation approach has been widely used for combining sources of information, such as in dynamic clustering [3] and semi-supervised learning [11].

1. Build spectral embedding
   - Construct the normalised feature-object matrix $\hat{\mathbf{A}}_t = \mathbf{D_1}^{-\frac{1}{2}}\mathbf{A}\mathbf{D_2}^{-\frac{1}{2}}$.
   - Compute $\mathbf{Z}_t$ from the truncated SVD of $\hat{\mathbf{A}}_t$ according to Eqn. 1.
   - Normalise the rows of $\mathbf{Z}_t$ to unit length.
2. Initialisation and prediction
   - If $t = 1$, apply orthogonal initialisation to select a set of k_t representative centroids from the representations of the objects in the embedded space.
   - For $t > 1$, recompute the k_{t-1} centroids based on last clustering but including only the embedding of the relevant set of objects/features in the current space.
   - If not all rows of the embedding have been assigned, apply nearest centroid classification to compute the predicted clustering $\mathcal{P}_t$.
3. Compute clustering
   - Apply constrained k-means to rows in $\mathbf{Z}_t$, initialised by centroids from $\mathcal{P}_t$.

Fig. 2. Dynamic spectral co-clustering process, as applied for each time step t.

The parameter $\alpha \in [0, 1]$ controls the balance between the influence of historical information and the information present in the current spectral embedding. A higher value of α allows information from the previous time step to have a greater influence, yielding a smoother transition between clusterings at successive time steps. Naturally at time $t = 1$, the right-hand term in Eqn. 4 will be zero.

Eqn. 4 can be viewed as the standard spherical k-means objective [12], augmented by a constraint reward term. We can find a local solution for this problem by using an approach analogous to the semi-supervised PCKMeans algorithm proposed by Basu *et al.* [11] for clustering with pairwise constraints. Specifically, we apply an iterative k-means-like assignment process, re-assigning each row vector z_i from $\mathbf{Z}_t$ to maximise:

$$\max_{C \in C_t} \; (1 - \alpha) \cdot z_i^{\intercal} \mu_c + \alpha \cdot \text{pred}(z_i, C) \tag{5}$$

where the quantity $pred(z_i, C)$ represents the degree to which the predicted assignment for the row z_i in P_t agrees with the assignment of z_i to cluster C. This is given by the proportion of rows in C that were co-assigned with z_i in P_t:

$$\text{pred}(z_i, C) = \frac{1}{|C|\,(|C| - 1)} \sum_{(z_i, z_j) \in C} co(z_i, z_j) \tag{6}$$

Once the algorithm has converged to a local solution, $\mathcal{C}_t$ provides us with a k-way partitioning of all nodes in the graph G_t (*i.e.* features and objects). An overview of the complete co-clustering process is shown in Figure 2.

3.3 Tracking Clusters Over Time

In the previous section we proposed an approach for co-clustering individual time step graphs. The second aspect of the offline approach to dynamic clustering

involves identifying *dynamic clusters* composed from clusters associated across time steps. We suggest that previous frameworks for tracking evolving dynamic communities [2, 13] can be readily adapted to the dynamic bipartite case. In brief, we construct a set of dynamic cluster timelines, each consisting of a set of clusters identified at different time steps and ordered by time. At each step in the dynamic co-clustering process, we match the predicted clusters (corresponding to clusters from the previous time step) with the actual output of the co-clustering process outlined in Figure 2. Matches are made based on the step cluster memberships for subsets of objects and/or features persisting between pairs of consecutive steps. This matching process will result in a set of dynamic clusters persisting across multiple steps.

4 Evaluation

4.1 Benchmark Evaluation

To evaluate the performance of the algorithm proposed in Section 3.2, we required an annotated dataset with temporal information. For this purpose we consider the bipartite document clustering problem, and use a subset of the widely-used Reuters RCV1 corpus [14]. The *RCV1-5topic* dataset[1] consists of 10,116 news articles covering a seven month period. Each article is annotated with a single ground truth topical label: health, religion, science, sport, weather. These topics are present across the entire time period of the corpus. We considered a number of different time step durations to split the seven month period – one month, a fortnight, and one week – yielding 7, 14, and 28 step graphs respectively. Naturally for this type of data, a subset of features (terms) will persist across time, while objects (documents) appear in only one time step.

Our evaluations focused on the performance of the dynamic spectral co-clustering algorithm on each time step graph in the *RCV1-5topic* dataset, using a range of values $\alpha \in [0.1, 0.5]$ for the balance parameter. As a baseline competitor, we used multi-partition spectral co-clustering as proposed by Dhillon [1]. To provide a fair comparison, we use orthogonal initialisation for both algorithms, and set the number of clusters k_t at time t to the number of ground truth topics.

Temporal smoothness. One of the primary motivations for dynamic co-clustering is to increase smoothness in the transitions between time step clusterings. To quantify the degree to which the proposed algorithm can enforce temporal smoothness, we measure the agreement between successive clusterings in terms of their *normalised mutual information* (NMI) [15]. Note that NMI values were calculated only over the terms common to each pair of consecutive time steps – documents are not considered as they do not persist.

Figure 3 shows a comparison of agreement values for the three different time window sizes. Dynamic co-clustering leads to a higher level of agreement than

[1] Datasets for this paper are available at `http://mlg.ucd.ie/datasets/dynak.html`

standard spectral co-clustering for all three time window sizes. The effect becomes significantly more pronounced as α increases. This is to be expected, as increasing the parameter leads to a higher weighting for the historic information in Eqn. 4. For $\alpha \geq 0.5$, the resulting co-clusterings are often almost identical to the predicted co-clustering $\mathcal{P}_t$, with the constrained k-means process converging to a solution after 2-5 iterations.

Clustering accuracy. To quantify algorithm accuracy, we calculated the NMI between clusterings and the relevant annotated document label information for

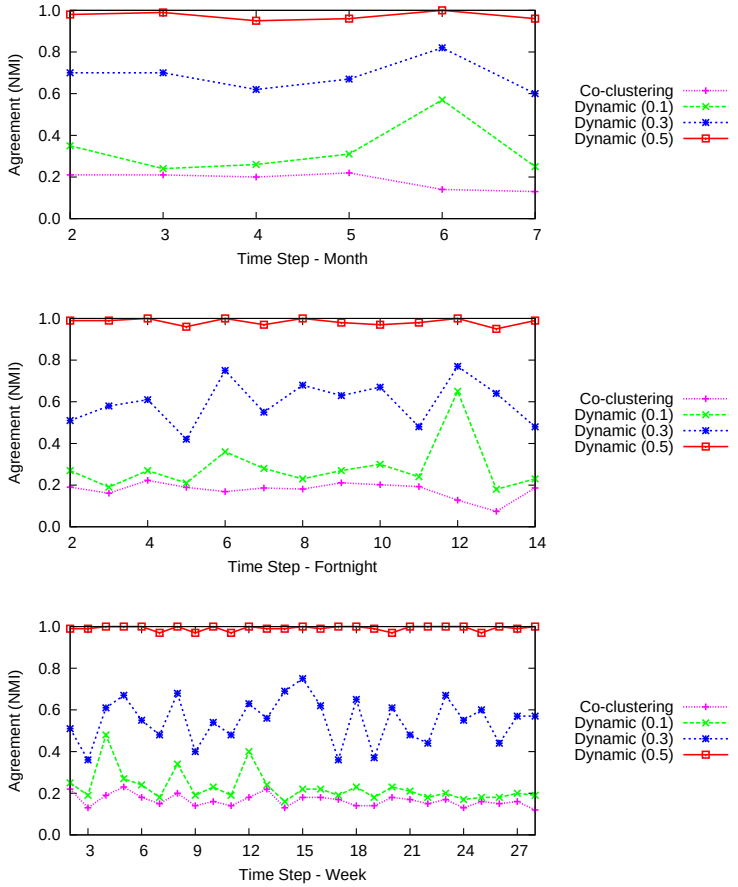


Fig. 3. Comparison of agreement (in terms of NMI) between successive feature clusterings, generated by spectral co-clustering and dynamic co-clustering ($\alpha \in [0.1, 0.5]$), on the *RCV1-5topic* dataset for monthly, fortnightly, and weekly time steps.

each time step. Figure 4 illustrates a comparison of the accuracy achieved by traditional spectral co-clustering and dynamic co-clustering on the *RCV1-5topic* dataset for the three different time step sizes. We observed that, for monthly and fortnightly time steps, the accuracy achieved by dynamic co-clustering was not significantly higher. However, for the weekly case, there was a noticeable increase in accuracy. In the case of $\alpha = 0.5$, dynamic co-clustering lead to higher accuracy on 21 out of 28 of the weekly graphs.

These results could appear surprising given the increases in temporal smoothness demonstrated Figure 3. However, on closer inspection, it is apparent that there is a strong *concept drift* effect in the data, as the composition of topics

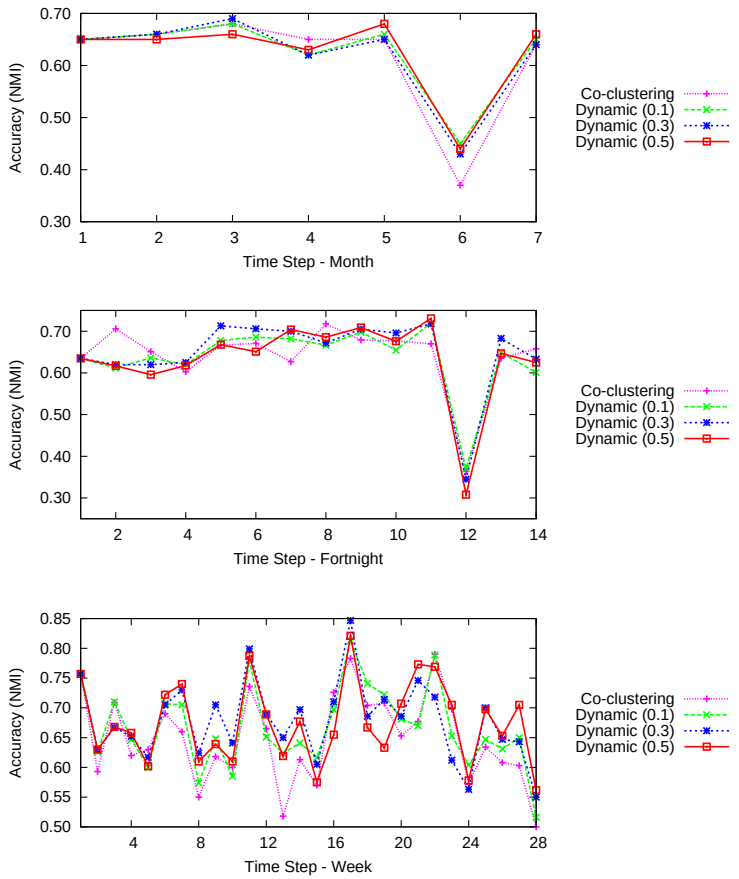


Fig. 4. Comparison of accuracy (in terms of NMI) for document clusterings generated by spectral co-clustering and dynamic co-clustering ($\alpha \in [0.1, 0.5]$), on the *RCV1-5topic* dataset for monthly, fortnightly, and weekly time steps.

changes over seven months. Therefore, for longer time periods, there is a greater change in the clusters identified in successive time periods. In such cases we expect historic information to be less useful. For the shorter weekly time windows, where there is less scope for drift between steps, we expect the use of historic information to improve accuracy. These results highlight the importance of selecting an appropriate time step size for offline dynamic clustering.

4.2 Evaluation on Web 2.0 Data

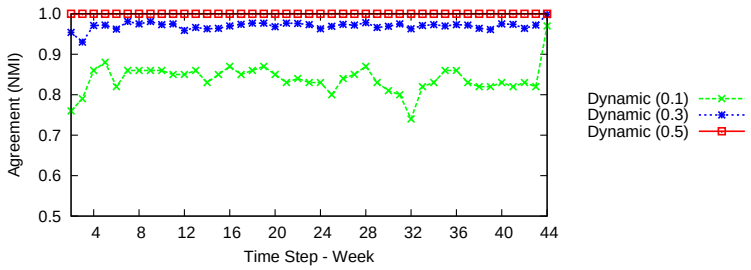
For the second phase of our evaluation, we applied the proposed co-clustering algorithm to a Web 2.0 data exploration problem. Unlike the RCV1 data, subsets of **both** objects (bookmarks) and features (tags) persist over time. We use a subset of the most recent data from a collection harvested by Görlitz *et al.* [16] from the *Del.icio.us* web bookmarking portal. The subset covers the 2000 top tags and 5000 top bookmarks across an eleven month period from January-November 2006. We divided this period into 44 weekly time steps, and for each time step we constructed a bipartite graph – the nodes represent tags and bookmarks, and the edges between them denote the number of times each bookmark was assigned a given tag during the time step. On average, each graph contained approximately 3750 bookmarks and 1760 tags. For each time step, we applied dynamic co-clustering for $k_t = 20$ to identify high-level topical clusters.

Figure 5 illustrates the agreement between both tag and bookmark clusterings identified by dynamic co-clustering for a balance parameter range $\alpha \in [0.1, 0.5]$. As with the *RCV1-topic* data, an increase in the value of α leads to clusters that are considerably more similar to those produced in the previous step, yielding smoother transitions between both feature and object clusters across time. In the extreme case of $\alpha = 0.5$, there is effectively no change between the predicted memberships and the final output of the co-clustering algorithm.

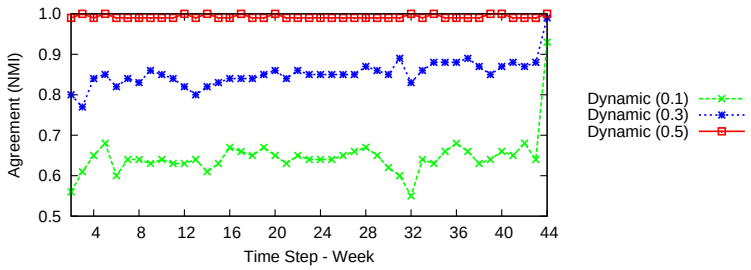
A number of authors (*e.g.* [17]) have suggested analysing the stability or "loyalty" of object member-cluster memberships across time. In the bipartite case, we can quantify this for both objects and features – we suggest the latter can be used to generate meaningful labels for dynamic clusters. For the tagged data, we score the fraction of time steps at which a tag is assigned to a given dynamic cluster. Over a sufficiently large number of steps, for each dynamic cluster we can produce a robust ranking of tags based on their respective membership stability scores. Examining the range of α parameters, we found the trade-off afforded by $\alpha = 0.1$ lead to the most interpretable label sets. In Table 1 we show the resulting descriptive labels selected for the dynamic clusters that exhibited the highest average tag membership stability, together with a suggested topic based on the tags. These descriptions highlight a range of general areas of interest covering sites frequently bookmarked by users during 2006.

5 Conclusion

In this work, we have described a spectral co-clustering algorithm for simultaneously clustering both objects and features in dynamic feature-based data,



(a) Object (Bookmark) Clustering Agreement



(b) Feature (Tag) Clustering Agreement

Fig. 5. Agreement between successive object and feature clusterings, identified by dynamic co-clustering ($\alpha \in [0.1, 0.5]$), on the *Del.icio.us* dataset across 44 weeks.

Topic	*Top 10 Tags*
IT	shortcuts, tweaks, opensource, security, troubleshooting, system, livecd, keyboard, sysadmin, ssh
Education	academic, school, mathematics, education, spanish, grammar, elearning, learning, math, slang
Music & Video	podcasts, mp3blog, youtube, television, movie, bittorrent, divx, torrent, p2p, npr
News & Media	newspapers, culture, opinion, society, iraq, news, journalism, environment, activism, political
Web Browsing	explorer, thunderbird, browser, opera, firefox, extensions, mozilla, greasemonkey, computing, plugin

Table 1. Top 10 tags for 5 most stable clusters (in terms of tag memberships over time) identified on the *Del.icio.us* dataset by dynamic co-clustering ($\alpha = 0.1$).

represented as a sequence of bipartite graphs. The co-clustering algorithm incorporates both current and historic information into the clustering process. A key aspect of the approach is that it is applicable in domains where objects or

features alone persist across time steps. In applications on both dynamic text and bookmark tagging data, the proposed approach was successful in identifying coherent clusters, while also ensuring a consistent transition between clusterings in successive time steps.

Acknowledgments. This work is supported by Science Foundation Ireland Grant No. 08/SRC/I140 (Clique: Graph & Network Analysis Cluster)

References

1. Dhillon, I.S.: Co-clustering documents and words using bipartite spectral graph partitioning. In: Proc. (2001) 269–274
2. Tantipathananandh, C., Berger-Wolf, T., Kempe, D.: A framework for community identification in dynamic social networks. In: Proc. 13th International conference on Knowledge Discovery and Data mining (KDD '07). (2007) 717–726
3. Chakrabarti, D., Kumar, R., Tomkins, A.: Evolutionary clustering. In: Proc. 12th Int. Conf. on Knowledge Discovery and Data Mining. (2006) 554–560
4. Mirzal, A., Furukawa, M.: Eigenvectors for clustering: Unipartite, bipartite, and directed graph cases. arXiv (2010)
5. Yang, Y., Pierce, T., Carbonell, J.: A study of retrospective and on-line event detection. In: Proc. 21st International ACM SIGIR Conference on Research and development in information retrieval. (1998) 28–36
6. Chi, Y., Song, X., Zhou, D., Hino, K., Tseng, B.: Evolutionary spectral clustering by incorporating temporal smoothness. In: Proc. 13th SIGKDD Int. Conf. on Knowledge Discovery and Data Mining. (2007) 153–162
7. Palla, G., Barabási, A., Vicsek, T.: Quantifying social group evolution. Nature **446**(7136) (2007) 664
8. Kalnis, P., Mamoulis, N., Bakiras, S.: On discovering moving clusters in spatio-temporal data. Proc. SSTD 2005 (2005) 364–381
9. Ng, A., Jordan, M., Weiss, Y.: On Spectral Clustering: Analysis and an Algorithm. Advances in Neural Information Processing **14**(2) (2001) 849–856
10. Tibshirani, R., Walther, G., Botstein, D., Brown, P.: Cluster validation by prediction strength. Technical report, Dept. Statistics, Stanford University (2001)
11. Basu, S., Banerjee, A., Mooney, R.: Active semi-supervision for pairwise constrained clustering. In: Proc. SIAM Int. Conf. on Data Mining. (2004) 333–344
12. Dhillon, I.S., Modha, D.S.: Concept decompositions for large sparse text data using clustering. Machine Learning **42**(1-2) (January 2001) 143–175
13. Greene, D., Doyle, D., Cunningham, P.: Tracking the evolution of communities in dynamic social networks. In: Proc. International Conference on Advances in Social Networks Analysis and Mining (ASONAM'10). (2010)
14. Lewis, D.D., Yang, Y., Rose, T.G., Li, F.: RCV1: A New Benchmark Collection for Text Categorization Research. JMLR **5** (2004) 361–397
15. Strehl, A., Ghosh, J.: Cluster ensembles - a knowledge reuse framework for combining multiple partitions. JMLR **3** (December 2002) 583–617
16. Görlitz, O., Sizov, S., Staab, S.: Pints: Peer-to-peer infrastructure for tagging systems. In: Proc. 7th International Workshop on Peer-to-Peer Systems. (2008)
17. Berger-Wolf, T., Saia, J.: A framework for analysis of dynamic social networks. In: Proc. 12th Int. Conf. on Knowledge Discovery and Data Mining. (2006) 523–528

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

Christian Bockermann and Felix Jungermann

Technical University of Dortmund, Artificial Intelligence Group

Abstract. The discovery of communities or interrelations in social networks has become an important area of research. The increasing amount of information available in these networks and its decreasing life-time poses tight constraints on the information processing – storage of the data is often prohibited due to its sheer volume.
In this paper we adapt a flexible approach for community discovery offering the integration of new information into the model. The continuous integration is combined with a time-based weighting of the data allowing for disposing obsolete information from the model building process.
We demonstrate the usefulness of our approach by applying it on the popular *Twitter* network. The proposed solution can be directly fed with streaming data from *Twitter*, providing an up-to-date community model.

1 Introduction

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

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

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

Fig. 1. Example tweets of the *Twitter* platform

A well-established representation of multi-dimensional relations is given by tensors [1, 2, 5, 13, 9, 6, 15]. A tensor is a multi-way array and can be seen as a generalization of a matrix. Tensors have been successfully used in multi-dimensional analysis and recently gained attention in social network mining [1, 2, 5]. In the case of social networks, tensors can be used to describe n-ary relations by using one tensor for each type of relations. Ternary relations of type (*user*,*tag*,*url*) can then be described by a mode-3 tensor $\boldsymbol{\mathcal{X}}$ with

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

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

Tensor based Community Discovery

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

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

Each of the matrices $\boldsymbol{U}^{(i)}$ in turn reflects a mapping of entities to clusters $\{1, \ldots, k\}$. The $[z]$ factor is a super-diagonal tensor which serves as a "glue element" – see Section 3 for details. A variety of different decomposition techniques such as Tucker3 or PARAFAC (CP) has been previously proposed [3, 10, 7]. Approximation is commonly measured by some divergence function. In [5] the authors proposed a clustering framework based on tensor decompositions which has been generalized for Bregman divergences. In [4] Bader et al. used CP tensor decomposition to detect and track informative discussions from the Enron email dataset by working on the ternary relation (*term*,*author*,*time*). These approaches have been applied to decompose single tensors. In [12] the authors introduced METAFAC, which is a factorization of a *set of tensors* with shared factors ($\boldsymbol{U}^{(q)}$ matrices). This allows for the discovery of one global clustering based on multiple tensor descriptions of the data. The time complexity for these tensor decompositions is generally given by the number of non-zero elements of the tensors (provided that a sparse representation is used).

Stream-based Community Discovery The majority of the tensor decomposition methods so far is based on a static data set. To incorporate streaming data, the stream is broken down into blocks and the decompositions are re-computed

for each of the new blocks [12]. A common way to handle time is to introduce a trade-off factor of the old data and the data contained in the new blocks.

In [14] Sun et al. presented dynamic tensor analysis. They handle n-ary relations by tensor decomposition using stream-based approximations of correlation matrices. They also presented a stream-based approach which is not really comparable to ours. They are processing a tensor containing data by unfolding the tensors to every single mode and after that they are handling every column of the resulting matrices in a stream to update their model. In reality, we cannot assume such an original tensor to be given. In contrast to [14], we consider multiple relations which have to be updated at each iteration instead of just one.

Contributions

The critical bottle-neck within the tensor decomposition methods often is their runtime. As of [12], the runtime for a decomposition of a set of tensors can be bound by $O(N)$, where N is the number of entries in all tensors. However, this number can be rather large – we extracted about 590k entries (relations) from 200k messages of the *Twitter* platform.

In this work, we present an adaption of the MetaFac framework proposed in [12]. Our contributions are as follows:

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

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

2 Multi-Relational Graphs

As denoted above, a social network generally consists of a set of related entities. In general, we are given sets $V_1, \ldots, V_k$ of entities of different types, such as

users, *keywords* or *urls*. Let V_i be the i-th type of entities, e.g. V_1 corresponds to *users*, V_2 refers to *keywords* and so on. A *relation* then is a tuple of entities, e.g. a *user-keyword* relation (u_1, k_1) is an element of $V_1 \times V_2$. We also refer to $R := V_1 \times V_2$ as the *relation type* R of the relation (u_1, k_1).

The entities are given as strings, and we define a mapping φ_i for each entity type V_i, which maps entities to integers $\varphi_i : V_i \to \{0, \ldots, |V_i| - 1\}$. The mapping φ_i can be some arbitrary bijective function. For some $w \in V_i$ we refer to $\varphi_i(w)$ as the index of w. We denote the string of an entity given by its index j by $\varphi^{-1}(j)$. This allows us to identify each entity by its index and enables us to describe a set of relations between entities by a tensor.

A tensor $\boldsymbol{\mathcal{X}}$ is a generalization of a matrix and can be seen as a higher-order matrix. A mode-k tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times \ldots \times I_k}$ is a schema with k dimensions where $\boldsymbol{\mathcal{X}}_{i_1,\ldots,i_k} \in \mathbb{R}, i_j \in I_j$ denotes the entry at position $(i_1, \ldots, i_k)$. For $k = 2$ this directly corresponds to a simple matrix whereas $k = 3$ is a cube.

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

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

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

2.1 MetaGraph

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

MetaGraph introduced by [12] is a relational hypergraph representing multi-dimensional data in a network of entities. A MetaGraph is defined as a graph $G = (V, E)$, where each vertex corresponds to a set of entities of the same type and each edge is defined as a *hyper-edge* connecting two or more vertices. By the use of hyper-edges, the MetaGraph captures multi-dimensional relations of the social network and therefore provides a framework to model n-ary relations.

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

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

2.2 Community Discovery Problem

With the use of tensors we have an approximated description of our social networks by means of a set of relation types $R_1, \ldots, R_n$. Thus we can describe our network graph G by means of the data tensors which are defined according to the observed relations $R_1, \ldots, R_n$ in G, i.e. $G \mapsto \left\{ \boldsymbol{\mathcal{X}}^{(1)}, \ldots, \boldsymbol{\mathcal{X}}^{(n)} \right\}$. Based on this description we seek for a further partitioning of the tensor representation into clusters of entities.

The solution proposed in [12] is a factorization of the tensors $\boldsymbol{\mathcal{X}}^{(i)}$ into products of matrices $\boldsymbol{U}^{(q)}$ which share a global factor $[z]$ and some of the $\boldsymbol{U}^{(q)}$ matrices. Let $\boldsymbol{\mathcal{X}}^{(i)}$ be the tensor describing $V_{i_1} \times \ldots \times V_{i_{l(i)}}$, then we can factorize this as

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

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

With an appropriate normalization as used in [12], the $\boldsymbol{U}^{(q)}$ matrices only contain values of $[0, 1]$ which can be interpreted as probability values. Based on this, the value of $\boldsymbol{U}^{(q)}_{l,m}$ can be seen as the probability of entity $\varphi_q^{-1}(l)$ belonging to cluster $m \in \{1, \ldots, k\}$ and we can simply map an entity to its cluster $C(l)$ by

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

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

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

3 MetaFac - Metagraph Factorization

As mentioned before, the Metagraph is a description of a multi-relational graph G by means of a set of tensors $\{\boldsymbol{\mathcal{X}}^{(i)}\}$. The objective of the MetaFac algorithm is to derive tensor decompositions of the $\boldsymbol{\mathcal{X}}^{(i)}$ with shared factors $[z], U^{(q)}$

which closely resemble the $\boldsymbol{\mathcal{X}}^{(i)}$. To measure the approximation, [12] proposed the Kullback Leibler divergence D_{KL}, thus implying the following optimization problem:

$$\underset{[z],\{U^{(q)}\}}{\arg\min} \sum_{i=1}^{n} D_{KL}(\boldsymbol{\mathcal{X}}^{(i)}, [\boldsymbol{z}] \prod_{i_1,\ldots,i_{l(i)}} \times_j \boldsymbol{U}^{(i_j)}) \tag{5}$$

To solve for (5) the authors derived an approximation scheme by defining

$$\boldsymbol{\mu}^{(i)} = \text{vec}(\boldsymbol{\mathcal{X}}^{(i)} \oslash ([\boldsymbol{z}] \prod_{j=1}^{l(i)} \times_j \boldsymbol{U}^{(i_j)})) \tag{6}$$

$$\boldsymbol{\mathcal{S}}^{(i)} = \text{fold}(\boldsymbol{\mu}^{(i)} * (\boldsymbol{z} * \boldsymbol{U}^{M_i} * \cdots * \boldsymbol{U}^{1_i})^T) \tag{7}$$

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

$$\boldsymbol{z} = \frac{1}{n} \sum_{i=1}^{n} \text{acc}(\boldsymbol{\mathcal{S}}^{(i)}, M_i + 1) \tag{8}$$

$$\boldsymbol{U}^{q} = \sum_{l:e_l \sim v_q} \text{acc}(\boldsymbol{\mathcal{S}}^{(i)}, q, M_e + 1) \tag{9}$$

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

The batched version of the MetaFac approximation can be derived by using the KL-divergence resulting in an appropriate approximation scheme, proposed by the following update function:

$$\boldsymbol{z} = (1-\alpha) \sum_{i=1}^{n} \text{acc}(\boldsymbol{\mathcal{S}}^{(i)}, M_i + 1) + \alpha \boldsymbol{z}_{t-1} \tag{10}$$

$$\boldsymbol{U}^{(q)} = (1-\alpha) \sum_{l:e_l \sim v_q} \text{acc}(\boldsymbol{\mathcal{S}}^{(j)}, q, M_i + 1) + \alpha \boldsymbol{U}^{(q)}_{t-1} \tag{11}$$

4 Stream-based Community Discovery with Tensors

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

Given a social network we are provided with a sequence M of messages $M := \langle m_0, m_1, \ldots \rangle$ where each message m_i implies a set of relations $\mathcal{R}(m_i)$. Let $\tau(m_i) \geq 0$ be the arrival time of m_i. This results in an overall sequence of

relations $S := \langle \mathcal{R}(m_0), \mathcal{R}(m_1), \ldots \rangle$ which are continuously added to the evolving social network graph G. Hence we are faced with a sequence $\langle G_{t_0}, G_{t_1}, \ldots \rangle$ of graphs where each G_{t_i} contains the relations of all messages up to time t_i.

Let t, t' be points in time with $t < t'$. In the following we will by $G_{[t,t']}$ denote the graph implied by only the messages of time-span $[t, t']$, hence $G_t = G_{[0,t]}$. Accordingly the graphs are represented by the corresponding tensor as $G_{[t,t']} \mapsto \left\{ \boldsymbol{\mathcal{X}}^{(1)}, \ldots, \boldsymbol{\mathcal{X}}^{(n)} \right\}_{[t,t']}$.

4.1 The MFstream Algorithm

The MetaFac approach uses a sliding window of some fixed window size w_s to manage streams. Given a sequence of time points t_j for $j \in \mathbb{N}$ with $t_j = t_{j-1} + w_s$, it factorizes $\{\boldsymbol{\mathcal{X}}^{(i)}\}_{[t_{j-1},t_j]}$ based on a trade-off factor α as denoted in equation (10) and (11).

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

4.2 Time-based Relation Weighting

So far we considered the property of two or more entities to be related as binary property, i.e. if entities i, j and k are related, then $\boldsymbol{\mathcal{X}}_{i,j,k} = w$, with $w \in \{0, 1\}$. With the extraction of relations from time-stamped messages – as provided within the *Twitter* platform – we are interested in incorporating the age of these relations to reflect the decreasing up-to-dateness of the information.

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

$$\boldsymbol{\mathcal{X}}^{(i)}_{i_1,\ldots,i_{l(i)}} = \begin{cases} \tau(r) & \text{if } \; r = (\varphi_1^{-1}(\nu_1), \ldots, \varphi_{l(i)}^{-1}(\nu_{l(i)})) \in S \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

In addition to that, we introduce a *global clock*, denoted by $\tau_{\max}$, which represents the largest (i.e. the most recent) timestamp of all relations observed so far, i.e. $\tau_{\max} := \max \{ \tau(r) \mid r \in X \}$. Storing the timestamp $\tau(r)$ for each entry r in the tensors allows us to define a weighting function for the relations based on the global clock value. A simple example for a parametrized weighting function is given as

$$\omega_{\alpha,\beta}(r) := \frac{\alpha}{\alpha + \frac{1}{\beta}(\tau_{\max} - \tau(r))}. \tag{13}$$

4.3 Sampling

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

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

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

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

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

4.4 Continuous Integration

With the prerequisites of section 4.2 and 4.3 we now present our stream adaption MFstream as Algorithm 1. MFstream is a purely dynamic approach of MetaFac which adds new relations to the data tensors $\{\boldsymbol{\mathcal{X}}^{(q)}\}$ and fits the model $[z], \{\boldsymbol{U}^{(q)}\}$ after a specified number of T messages. This differentiates our approach from MetaFac as the optimization is performed by running a single iteration of the optimization loop – with respect to the time-based weighting – after adding the relations of T messages to the tensors. The time complexity per iteration of the MFstream algorithm is the same as for the MetaFac algorithms (see Section 3). Due to the fixed tensor dimensions, the maximum number of non-zero elements N is constant, which implies $O(1)$ runtime.

5 Evaluation

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

To discover clusters on the above mentioned entities present on the *Twitter* platform, we constructed a metagraph for *Twitter*. The entity types themselves

Algorithm 1 The MFSTREAM algorithm.

Input: MetaGraph $G = (V, E)$, Stream $M = \langle m_i \rangle$, capacities C_q, number of clusters k, constant $T \in \mathbb{N}$
procedure MFSTREAM
  Initialize z, $\{\boldsymbol{U}^{(q)}\}$, $c := 0$
  while $M \neq \emptyset$ **do**
    $m := m_c$, $c := c + 1$ ▷ *Pick the next message from the stream*
    for all $(r_{j_1}, \ldots, r_{j_{l(j)}}) \in \mathcal{R}(m)$ **do**
      for all $p = 1, \ldots, l(j)$ **do**
        if $\varphi_p(r_{j_p}) = nil$ **then** ▷ *Replacement needed?*
          if $|\varphi_p| = C_q + 1$ **then**
            $f^* := \arg\min_{f \in \varphi_p} \tau(f)$
            $\boldsymbol{U}^{(p)}_{f^*,s} := \frac{1}{k} \; \forall \, s = 1, \ldots, k$
          else
            $f^* := \min_{f \in \{1,\ldots,C_p\}} \varphi_p^{-1}(f) = nil$ ▷ *Pick next unmapped* f^*
          end if
          $\varphi_p(r_{j_p}) := f^*$, $\tau(f^*) := \tau(m)$
        end if
      end for
    end for
    $\nu_i := \varphi_p(r_{j_i})$ for $i = 1, \ldots, l(j)$
    $\boldsymbol{\mathcal{X}}^{(p)}_{\nu_1,\ldots,\nu_{l(j)}} := \tau(m)$ ▷ *Update corresponding tensor*
    if $c \equiv 0 \mod T$ **then** ▷ *Single opt.-iteration every* T *steps*
      for all $i \in \{1, \ldots, n\}$ **do**
        compute $\{\boldsymbol{\mathcal{S}}^{(i)}\}$ by eq. (7) and (6)
        update z by eq. (8)
      end for
      for all $j \in \{1, \ldots, q\}$ **do**
        update $\{\boldsymbol{U}^{(j)}\}$ by eq. (9)
      end for
    end if
  end while
end procedure

imply as much as $\mathcal{P}(V) = 2^4$ possible relation types, some of which will not arise or are redundant. E.g. since each *tweet* is written by a user, there is no relation (*tweet*,*tag*) which does not also refer to a *user*. Hence, our MetaGraph is based on the relation types $\{R_1, \ldots, R_8\}$ given in Fig. 2. We extracted 1000 seed

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

Fig. 2. Relation types for the *Twitter* metagraph

users and their direct *friends* and *followers*. *Followers* are following a *user* which means that messages of the *user* are directly visible for the *followers* at their *twitter* website. *Friends* are all the users a particular *user* is following. We used an English stopword filter to extract users which are writing in English language

and processed all *friends* and *followers* of the seed users, revealing about 478.000 *users*. For these, we extracted all the messages written between the 19th and 23rd of February 2010. Out of these 2.274.000 *tweets* we used the *tweets* written at the 19th of February for our experiments, leaving about 389.000 *tweets* from 41.000 *users*.

5.1 Evaluating the Model

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

$$W(C) := \sum_{k=1}^{K} N_k \sum_{C(i)=k} \parallel x_i - \overline{x}_k \parallel^2 \tag{14}$$

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

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

To be able to compare the clusterings of MFSTREAM and METAFAC, we processed messages until the first entity type V_i reached its limit and stored the resulting clustering on disk. Then we reset the φ mappings and started anew, revealing a new clustering every time an entity type i reached C_i, revealing a total of 93 clusterings. We applied METAFAC on the messages that have been used to create these 93 clusterings and computed their similarities using $W(C)$. Figure 5 shows that MFSTREAM delivers results comparable to METAFAC for different weighting functions. Figure 5 shows that using timestamped values instead of binary values for calculation of the MFSTREAM delivers better results. The decrease of T, which implies a larger number of optimization steps, intuitively increases the quality of MFSTREAM as is attested by Figures 4 and 6.

In addition, we made experiments to show the effect of the update frequency T on the runtime. Figure 7 shows the relative runtime of MFSTREAM where $T = 1$ corresponds to the baseline at 1.0. Raising T results in shorter runtime, since the model is updated less frequently, which is the major time factor. The upper curve shows the runtime for updating after 5 relations ($T = 5$), the middle one shows $T = 10$, and the latter refers to $T = 50$.

Varying sizes of entity types by the C_q results in clusterings of different numbers of entities, which cannot be directly compared by $W(C)$. Hence, we normalized $W(C)$ by the variance $\mathcal{V}$ of each clustering. Larger models of course incorporate more information, which results in more stable clusterings as can be seen in Figure 8.

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

Fig. 3. Mean $W(C)$ of different weights ($T = 20$), comparing MFStream and MetaFac

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

Fig. 4. Mean of $W(C)$ with different update steppings T (weight used: $\omega_{1,1000}$)

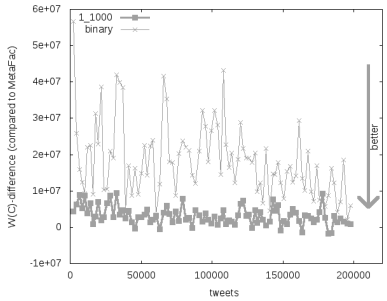


Fig. 5. $W(C)$ for MFStream compared to the MetaFac clusterings.

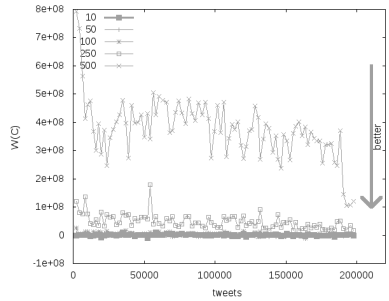


Fig. 6. Relative $W(C)$ of MFStream using different update step sizes T

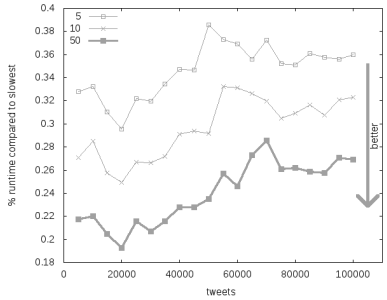


Fig. 7. Rel. runtime of MFStream using different numbers of relations for update.

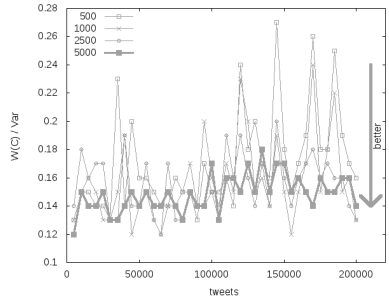


Fig. 8. $W(C)/\mathcal{V}$ of MFStream using different sizes of models.

6 Conclusion and Future Work

In this work we presented MFStream, a flexible algorithm for clustering multi-relational data from evolving networks, derived from the MetaFac framework by [12]. The main improvement of our approach is the reduction of the approximation scheme on to a small relevant window of relations. The proposed time-based weighting of relations contributes to this reduction by removing obsolete information that is not relevant to the model adaption anymore. MFStream is able to handle relations containing new, unseen entities by offering a replacement strategy for the set of entities considered at optimization time. This makes it

especially suitable to continuously integrate new data from a stream. We evaluated MFStream on real-world data crawled from the *Twitter* platform and showed its comparability to MetaFac.

The use of backend storage for off-loading obsolete data that can be re-imported into the optimization window at a later stage might be an interesting advancement. Also, concurrent criteria *runtime* and *quality* offer a starting point for multi-objective optimization. Additionally, recent works [11] motivate further improvements to handle a dynamic number k of clusters within MFStream.

References

1. E. Acar, S. A. Çamtepe, M. S. Krishnamoorthy, and B. Yener. Modeling and multiway analysis of chatroom tensors. In *ISI*, pages 256–268, 2005.
2. E. Acar, S. A. Çamtepe, and B. Yener. Collective sampling and analysis of high order tensors for chatroom communications. In *ISI*, pages 213–224, 2006.
3. J. F. Allen. Towards a general theory of action and time. *Artificial Intelligence*, 23:123–154, 1984.
4. B. Bader, M. W. Berry, and M. Browne. *Survey of Text Mining II*, chapter Discussion tracking in Enron email using PARAFAC, pages 147–163. Springer, 2007.
5. A. Banerjee, S. Basu, and S. Merugu. Multi-way clustering on relation graphs. In *SDM*. SIAM, 2007.
6. D. Cai, X. He, and J. Han. Tensor space model for document analysis. In E. N. Efthimiadis, S. T. Dumais, D. Hawking, and K. Järvelin, editors, *SIGIR*, pages 625–626. ACM, 2006.
7. R. Harshman. Foundations of the parafac procedure: Models and conditions for an "explanatory" multi-modal factor analysis. *UCLA Working Papers in Phonetics*, 16, 1970.
8. T. Hastie, R. Tibshirani, and F. J. The elements of statistical learning-data mining, inference and prediction. *Springer, Berlin Heidelberg New York*, 2001.
9. T. G. Kolda, B. W. Bader, and J. P. Kenny. Higher-order web link analysis using multilinear algebra. In *ICDM '05: Proceedings of the Fifth IEEE International Conference on Data Mining*, pages 242–249, 2005.
10. L. D. Lathauwer, B. D. Moor, and J. Vandewalle. A multilinear singular value decomposition. *SIAM J. Matrix Anal. Appl.*, 21(4):1253–1278, 2000.
11. Y.-R. Lin, J. Sun, N. Cao, and S. Liu. Contextour: Contextual contour visual analysis on dynamic multi-relational clustering. In *Proceedings of the SIAM Conference on Data Mining (SDM10)*, 2010. Not published, yet.
12. Y.-R. Lin, J. Sun, P. Castro, R. Konuru, H. Sundaram, and A. Kelliher. Metafac: Community discovery via relational hypergraph factorization. In *Proceedings of the SIGKDD 2009*, pages 527–536, Paris, France, 2009. ACM.
13. A. Shashua and T. Hazan. Non-negative tensor factorization with applications to statistics and computer vision. In *ICML '05: Proceedings of the 22nd international conference on Machine learning*, pages 792–799, New York, NY, USA, 2005. ACM.
14. J. Sun, D. Tao, and C. Faloutsos. Beyond streams and graphs: dynamic tensor analysis. In *Proceedings of the SIGKDD 2006*, pages 374–383, New York, NY, USA, 2006. ACM.
15. X. Wang, J.-T. Sun, Z. Chen, and C. Zhai. Latent semantic analysis for multiple-type interrelated data objects. In *Proceedings of the SIGIR 2006*, pages 236–243, New York, NY, USA, 2006. ACM.

Network-Based Disease Candidate Gene Prioritization: Towards Global Diffusion in Heterogeneous Association Networks

Joana P. Gonçalves[1,2,3,*], Sara C. Madeira[1,2], and Yves Moreau[3]

[1] Knowledge Discovery and Bioinformatics group (KDBIO), INESC-ID, Rua Alves Redol 9, 1000-029 Lisboa, Portugal
[2] IST, Technical University of Lisbon, Av. Rovisco Pais 1, 1049-001 Lisboa, Portugal
[3] BIOI, ESAT-SCD, Department of Electrical Engineering, KULeuven, Kasteelpark Arenberg 10, 3001 Leuven-Heverlee, Belgium
{jpg,smadeira}@kdbio.inesc-id.pt, yves.moreau@esat.kuleuven.be,

Abstract. Disease candidate gene prioritization addresses the association of novel genes with disease susceptibility or progression. Network-based approaches explore the connectivity properties of biological networks to compute an association score between candidate and disease-related genes. Although several methods have been proposed to date, a number of concerns arise: (i) most networks used rely exclusively on curated physical interactions, resulting in poor coverage of the Human genome and leading to sparsity issues; (ii) most methods fail to incorporate interaction confidence weights; (iii) in some cases, relevance scores are computed as local measures based on the direct interactions with the disease-related genes, ignoring potentially relevant indirect interactions. In this study, we seek a robust network-based strategy by evaluating the performance of selected prioritization strategies using genes known to be involved in 29 different diseases.

Keywords: protein-protein interaction, network, random walk, disease candidate genes, prioritization

1 Introduction

Biomarkers play a crucial role in modern medical practice as a means of improving accuracy in diagnosis, prognosis and treatment. In particular, research has been actively devising associations of novel genes with disease susceptibility or progression, relying on high-throughput technologies and the proliferation of accessible resources of biological data to enable large-scale genome-wide studies.

Most computational methods proposed for disease gene prioritization aim to identify putative candidates based on their similarity with genes known to be involved in the occurrence of a particular phenotype, according to: intrinsic properties, functional annotations, coherent transcriptional responses via expression data analysis, orthologous relations with genes from model organisms or even

co-occurrence in the literature [22]. Alternative strategies adopt a systemic approach and explore the topology of biological networks, including protein-protein interactions, regulatory data or metabolic pathways. These approaches rely on the assumption that genes co-occurring in a particular network substructure or interacting tend to participate together in related biological processes to identify novel genes based on their linkage with the known disease genes [22].

Integrative network-based analysis has been addressed [8,11,15,16,20,23,26], combining knowledge from distinct resources in association networks to unravel novel disease genes. However, most of these approaches rely solely on physical interactions [8,23], potentially inferred via orthologous relations with model organisms [26], often resulting in insufficient coverage of the Human genome. Others include additional interactions predicted from coexpression, pathway, functional or literature data, but still devise sparse networks [11,15]. Although the risk for false positive interactions may rise, the integration of knowledge from heterogeneous sources generates denser networks which tend to be less biased toward a particular evidence, more robust to noise and thus able to perform better in the prioritization task [16].

Network-based prioritization methods further differ in how they define the ranking of the candidates from the known disease-related genes. Local measures are usually computed based on the direct links or shortest paths between the candidates and the disease-related genes [15, 16], while global strategies diffuse or smooth a disease-related signal through the network. In this work, we evaluate whether the latter should be preferred over the former, as the inclusion of indirect associations is able to compensate for missing linkage, ultimately mitigating sparsity and "small world" effect issues [20], and global similarities have recently been shown to outperform local measures [15].

Random walks or diffusion kernels arise as natural candidates for the diffusion approach and their application to prioritization has been proven effective [6, 8, 15, 23]. Not only they compute fast using iterative methods, even for large networks [6], they are also able to straightforwardly establish a ranking of the candidates based on the global connectivity of the network. Nevertheless, some of the proposed methods [8, 15] ignore or fail to incorporate weights expressing the confidence on the evidence of every particular association [16]. Furthermore, their scores are based on the steady-state probability obtained after a large number of iterations or upon convergence. In this study, we assess the claim that limited diffusion is usually sufficient for ranking purposes [9, 10] and on our intuition which leads us to expect the prior knowledge to be somehow lost or of very little importance to the ranking after diffusing to a large extent.

Throughout this paper, we address the aforementioned topics by analyzing the performance of different prioritization strategies in three case studies: (i) Integrative heterogeneous protein association network *vs* integrative protein-protein physical interaction network (PPPIN); (ii) Global ranking measure *vs* local ranking measure; (iii) Confidence weights, degree of diffusion and parameter variation.

2 Methods

A protein-protein association network can be described as a weighted undirected graph, a special case of a weighted directed graph, defined as $G = (V, E)$, where V is the set of vertices and E is the set of edges. Each vertex in V and edge in E correspond to a gene and an association between two genes, respectively. Let A and D denote the adjacency and diagonal matrices of G, respectively. A_{uv} is the weight $w(u, v)$ of the edge (u, v) between source u and target v. Also, $D_{uu} = \sum_{(u,v)\in E} A_{uv}, \forall u \in V$, that is, the sum of the weights of the edges for which u is the source. Prioritizing disease candidates thus formulates as obtaining a ranking on V given a set $S \in V$ of seed genes. For the local scoring scheme Endeavour's measure was used [2]. As global network-based strategies, the PageRank with priors and Heat Diffusion random walks were applied: an initial signal expressing the relevance of the genes in the context of the disease in the form of a preference vector, $p^{(0)}$, is diffused over the network by performing a limited number of iterations, N.

2.1 Endeavour's Measure: Intersection of Interactors

Endeavour computes a local network-based measure, whereby the score of each gene is computed as the overlap between the sets of genes interacting with the seed genes and those interacting with the candidate gene itself [2]:

$$S_v = \sum_{(u,v)\in E} intSeeds(u) \qquad intSeeds(u) = \begin{cases} 1 & \text{, if } \exists z \in S : (u, z) \in E \\ 0 & \text{, otherwise} \end{cases}$$

2.2 Heat Diffusion and PageRank with Priors

Heat Diffusion is a discrete approximation of the heat kernel [28] first introduced in [9], in which the rate of diffusion is controlled by a non-negative parameter, the heat diffusion coefficient t. The iterative equation is given by

$$p_v^{(i+1)} = \left(1 - \frac{t}{N}\right) \cdot p_v^{(i)} + \frac{t}{N} \sum_{(u,v)\in E} p_u^{(i)} \cdot \frac{A_{uv}}{D_{uu}}.$$

PageRank with priors is an extension of the original PageRank algorithm to consider the original probability distribution of the scores [25]. A parameter β, called "back probability" expresses the probability of jumping to the initial node at each iteration. The iterative equation is

$$p_v^{(i+1)} = \beta \cdot p_v^{(0)} + (1 - \beta) \cdot \sum_{(u,v)\in E} p_u^{(i)} \frac{A_{uv}}{D_{uu}}.$$

3 Results

Evaluation sudies were performed using Human data from the STRING database [12] and a PPPIN from Entrez Gene [1] as representatives of protein-protein heterogeneous association and physical interaction networks, respectively. 620 genes known to be related with 29 diseases were used as prior knowledge to prioritize candidates in a leave-one-out cross-validation scheme.

3.1 Data and Preprocessing

Networks The STRING database [12, 18] integrates physical interactions and predicted associations based on knowledge obtained from heterogeneous sources of transcriptional, functional, metabolic, literature and orthology data. For a fair comparison with Endeavour, we downloaded and parsed version 7.1 of STRING [18], including evidences from MINT [7], HPRD [19], BIND [4], DIP [27], BioGRID [5], KEGG [13] and Reactome [24] databases. Associations from STRING v8.2 [12] were also retrieved to assess to which extent the additional knowledge integrated from IntAct [14], PID [21] and GO [3] protein complexes would improve the prioritization performance relative to the previous release. A PPPIN was downloaded from the NCBI Entrez Gene FTP repository [1]. 130797 Human interactions were selected from 448534 entries, for which both interactant genes were tagged with tax ID 9606. From these, 4611, 51275 and 74911 were originally from BIND [4], BioGRID [5] and HPRD [19], respectively. Genes' identifiers followed Entrez Gene nomenclature. Preprocessing of these networks involved filtering redundant edges and devising an explicit representation of a directed graph. In the case of the STRING releases, original weights were used to express the confidence of every association, while in the PPPIN all edges were attributed weight 1. STRING v7.1 contained 16050 genes and 698534 unique associations. STRING v8.2 covered 17448 Human genes with 1256016 non-redundant associations. Finally, the PPPIN had 47873 physical interactions between 10175 genes.

Seed sets 620 disease genes were selected from the OMIM [17] database spanning 29 disease-specific sets, with an average of 21 genes per set. As genes were identified according to Ensembl nomenclature, the seeds could be directly used with STRING. For the PPPIN, however, we performed a conversion between Ensembl and Entrez Gene identifiers. A mapping was parsed from a file downloaded from the NCBI Entrez Gene FTP repository [1] and used to generate the corresponding seed sets using Entrez Gene names. Additionally, we filtered the genes absent from at least one of the networks or for which the conversion between Ensembl and Entrez did not succeed. In total, 94 seed genes were lost (14 with no conversion, 80 absent from the PPPIN). A single occurrence of a gene with several Entrez aliases happened. In this case, only the alias present in the PPPIN was kept. For validation purposes, seed sets containing randomly selected genes were generated. The number of seeds in each set was randomly chosen in the range $[5, 100]$ and the genes were randomly selected from the Human STRING v8.2 network. 546 genes were retrieved.

3.2 Evaluation Measures and Experimental Setting

Evaluation measures Ideally, in a leave-one-out cross-validation scheme, we would expect the prioritization strategy to rank the left-out gene known to be related with the disease at the top. Under this assumption, we assess the performance of the scoring methods overall and per disease based on four evaluation measures: the number of left-out genes ranked in the top 10 and 20 positions, the Area under the ROC curve (AUC) score, and the mean average precision.

For a given combination of diffusion parameter α and number of iterations N, n rankings are generated (one per left-out gene). The AUC score is given by

$$S_{AUC_{\alpha,N}} = \frac{n - \sum_{k=1}^{n} \frac{r_k^{(N)}}{m_k^{(N)}}}{n},$$

where $r_k^{(N)}$ is the ranking position of the k^{th} left-out gene in the k^{th} ranked list and $m_k^{(N)}$ is the number of ranked genes in the k^{th} list.

Mean average precision (MAP) is an evaluation measure that combines precision and recall. Essentially, MAP averages the precisions computed by truncating the list after each of the relevant entities is found. Only one relevant entity must be found, the left-out gene. Thus, precision at rank r is either 0, before it has been found, or $\frac{1}{r}$. Moreover, in our setting the ranked lists contain equal number of genes, allowing us to simplify our MAP score for n lists with the same size to:

$$S_{MAP_{\alpha,N}} = \frac{\sum_{k=1}^{n} \frac{1}{r_k^{(N)}}}{n}$$

Experimental setting In each validation run, one different gene was deleted from the set of seed genes and added to 99 randomly selected candidate genes. A ranking method was then applied to compute a score for every gene in the network. Finally, the ranking of the 100 candidate genes was defined according to the retrieved scores. In the case of the Heat Diffusion, the scores of the seed genes were initialized to 1. For PageRank, an initial seed score of $1/|S|$ was used. Performance was assessed by computing AUC and MAP scores, and counting the number of left-out genes ranked in the top 10 and top 20 positions, both overall and per disease. We sought the best performance of each method using several combinations of parameters. Heat diffusion coefficients t and back probabilities β of 0.1, 0.3, 0.5, 0.7 and 0.9 with 2, 5, 10, 15 and 20 iterations using STRING and 2, 5, 10, 20, 100 iterations using the PPPIN were tried. In the case studies, results are shown only for the parameter settings which achieved the best performance in each case. We further ranked the randomly generated seed sets using the leave-one-out cross-validation in STRING v8.2 to assess whether the Heat Diffusion method was able to take advantage of the information contained in the seed sets to improve the identification of the left-out seeds. Overall, AUC and MAP scores of 0.501 and 0.05 were achieved and only 57 and 92 genes were ranked in the top 10 and 20 positions. Similar results were obtained per seed set (data not shown), in accordance with what would have been expected for random seed sets.

3.3 Case Studies

Heat Diffusion and PageRank with priors achieved similar results in both networks (Table 1). For this reason, we abstain ourselves of comparing the results of both random walks, considering the results equivalent when applied to the same network. Throughout this section, we will always refer to one of them as a representative of a global measure. A brief description of the prioritization performances obtained for each case study follows.

Method	Network	Parameters	AUC	MAP	TOP 10	TOP 20	#BRM	#BRN
HeatDiffusion	STRING8	$t = 0.3, N = 10$	0.962	0.711	484	502	26%	68%
PageRank	STRING8	$\beta = 0.7, N = 2$	0.961	0.693	485	502	20%	69%
HeatDiffusion	PPPIN	$t = 0.5, N = 2$	0.862	0.352	301	373	40%	11%
PageRank	PPPIN	$\beta = 0.5, N = 2$	0.861	0.349	304	384	38%	10%

Table 1. Results of Heat Diffusion and PageRank using both STRING v8.2 and the PPPIN. '#BRM' (better ranked by method in each network) shows the percentage of genes with a higher rank in a one-to-one comparison of the ranks per gene for both methods in each network. '#BRN' (better ranked by network for each method) shows the percentage of genes with a higher rank in a one-to-one comparison of the ranks per gene for both networks using the same method. Total number of genes: 526.

Global measure *vs* Local measure A network-based global ranking was obtained using the Heat Diffusion method with $t = 0.3, N = 10$, while Endeavour [2] was used to score the genes using its local measure. Both rankings were based on STRING v7.1, the version included in Endeavour. Overall, the random walk global measure outperformed the local interaction overlap in all evaluation measures (see Table 2), that is, the higher number of left-out genes was ranked on the top positions, also achieving better ranks in general, using the latter.

Method	Network	AUC	MAP	TOP 10	TOP 20
HeatDiffusion ($t = 0.3, N = 10$)	STRING v7.1	0.942	0.643	536	569
Endeavour	STRING v7.1	0.806	0.326	393	464

Table 2. Overall results of Heat Diffusion and Endeavour using STRING v7.1. Total number of genes: 620.

Regarding the AUC scores per disease (see Table 3), the Heat Diffusion method outperformed Endeavour in all diseases except Ehlers-Danlos syndrome (0.944 opposed to 0.948, respectively). This was also the only disease for which

the number of genes ranked in the top 20 positions was higher using the local measure (Endeavour was able to rank one more gene in the top 20). However, the MAP score was better for the Heat Diffusion method and, in fact, 9 of the 10 seed genes ranked in the top 10 positions by both methods scored higher using the global measure.

For the remaining diseases, Heat Diffusion was always able to rank the same or a higher number of genes in both the top 10 and the top 20 positions. Regarding the MAP scores, Heat Diffusion outperformed Endeavour in every disease and was able to rank all genes of both amyotrophic lateral sclerosis and Usher syndrome in the first position.

		Heat Diffusion STRING v7.1				Endeavour STRING v7.1			
Disease	#Genes	AUC	MAP	Top10	Top20	AUC	MAP	Top10	Top20
Alzheimer's disease	8	0.934	0.586	7	7	0.930	0.376	6	7
amyotrophic lateral sclerosis	4	0.990	1.000	4	4	0.975	0.550	4	4
anemia	44	0.928	0.499	36	40	0.718	0.187	21	30
breast cancer	24	0.930	0.608	21	22	0.782	0.214	13	19
cardiomyopathy	22	0.973	0.812	21	21	0.862	0.579	18	18
cataract	20	0.890	0.693	15	16	0.883	0.363	13	17
Charcot-Marie-Tooth disease	14	0.889	0.752	12	12	0.738	0.361	8	8
colorectal cancer	21	0.961	0.697	19	20	0.918	0.389	17	20
deafness	42	0.941	0.642	37	40	0.732	0.186	17	25
diabetes	26	0.967	0.731	22	26	0.820	0.232	17	21
dystonia	5	0.986	0.867	5	5	0.938	0.381	4	5
Ehlers-Danlos syndrome	10	0.944	0.650	9	9	0.948	0.296	9	10
emolytic anemia	13	0.965	0.683	12	12	0.737	0.269	8	8
epilepsy	15	0.989	0.933	15	15	0.749	0.612	10	10
ichthyosis	9	0.881	0.598	8	8	0.778	0.226	6	6
leukemia	112	0.922	0.428	88	100	0.807	0.203	68	86
lymphoma	31	0.920	0.420	24	25	0.796	0.275	19	22
mental retardation	24	0.918	0.629	21	21	0.624	0.110	7	11
muscular dystrophy	24	0.981	0.780	24	24	0.869	0.390	19	21
myopathy	41	0.961	0.594	37	39	0.885	0.535	34	34
neuropathy	18	0.965	0.671	14	17	0.648	0.205	8	9
obesity	13	0.931	0.796	12	12	0.918	0.559	12	12
Parkinson's disease	9	0.903	0.728	7	7	0.661	0.158	4	4
retinitis pigmentosa	30	0.957	0.882	27	28	0.845	0.470	22	23
spastic paraplegia	7	0.930	0.860	6	6	0.927	0.586	5	6
spinocerebellar ataxia	7	0.959	0.863	6	6	0.816	0.250	3	6
Usher syndrome	8	0.990	1.000	8	8	0.988	0.917	8	8
xeroderma pigmentosum	10	0.987	0.850	10	10	0.785	0.704	7	7
Zellweger syndrome	9	0.989	0.944	9	9	0.823	0.513	6	7

Table 3. Results of the Heat Diffusion ($t = 0.3$, 10 iterations) and Endeavour methods using STRING v7.1, per disease. Total number of genes: 620.

Protein-Protein Associations *vs* Protein-Protein Physical Interactions Heat Diffusion achieved better performance using STRING v8.2, with AUC score 0.962, opposed to 0.862 using the PPPIN (see Table 1). Furthermore, STRING enabled to rank more than 90% of the genes in the top 10 positions, while using the PPPIN less than 60% were in top 10. In a one-to-one comparison, Heat Diffusion ranked 68% of the genes better using STRING, while only 11% of the ranks were better using the PPPIN. Table 4 compares the results obtained for the Heat Diffusion method using STRING v8.2 with PageRank with priors in a PPPIN, one of the best performing strategies in [8], per disease.

		Heat Diffusion STRING v8.2				PageRank PPPIN			
Disease	#Genes	AUC	MAP	Top10	Top20	AUC	MAP	Top10	Top20
Alzheimer's disease	8	0.929	0.877	7	7	0.668	0.456	5	5
amyotrophic lateral sclerosis	4	0.990	1.000	4	4	0.530	0.028	0	1
anemia	37	0.967	0.599	35	36	0.679	0.268	15	19
breast cancer	22	0.952	0.618	20	20	0.877	0.427	17	18
cardiomyopathy	19	0.986	0.904	19	19	0.789	0.383	12	13
cataract	16	0.980	0.781	16	16	0.751	0.485	10	11
Charcot-Marie-Tooth disease	10	0.934	0.735	9	9	0.665	0.251	3	3
colorectal cancer	29	0.969	0.785	19	19	0.912	0.382	15	19
deafness	28	0.950	0.623	23	27	0.547	0.210	7	8
diabetes	25	0.966	0.743	23	24	0.838	0.422	17	20
dystonia	5	0.986	0.800	5	5	0.700	0.316	2	2
Ehlers-Danlos syndrome	8	0.990	1.000	8	8	0.850	0.613	6	7
emolytic anemia	12	0.978	0.772	12	12	0.793	0.149	4	6
epilepsy	13	0.989	0.962	13	13	0.803	0.454	8	8
ichthyosis	7	0.954	0.768	6	6	0.651	0.367	3	3
leukemia	98	0.948	0.520	86	93	0.811	0.209	50	67
lymphoma	26	0.930	0.476	21	22	0.850	0.270	15	18
mental retardation	19	0.926	0.727	16	17	0.739	0.303	8	12
muscular dystrophy	20	0.983	0.790	20	20	0.893	0.524	15	15
myopathy	35	0.969	0.702	33	35	0.731	0.272	20	24
neuropathy	17	0.951	0.699	15	15	0.636	0.201	5	8
obesity	12	0.988	0.917	12	12	0.892	0.621	10	10
Parkinson's disease	8	0.935	0.878	7	7	0.754	0.465	5	5
retinitis pigmentosa	23	0.981	0.883	22	23	0.736	0.310	11	12
spastic paraplegia	5	0.990	1.000	5	5	0.490	0.083	1	1
spinocerebellar ataxia	7	0.957	0.768	6	6	0.726	0.095	3	4
Usher syndrome	4	0.990	1.000	4	4	0.880	0.631	3	3
xeroderma pigmentosum	10	0.988	0.900	10	10	0.980	0.811	10	10
Zellweger syndrome	8	0.990	1.000	8	8	0.871	0.814	7	7

Table 4. Heat Diffusion using STRING v8.2 ($t = 0.3$, $N = 10$) vs PageRank with priors using the PPPIN ($\beta = 0.5$, $N = 2$), per disease. Total number of genes: 526.

Regarding the disease-specific scores (see Table 4), the lowest AUC (and MAP) values for the combination Heat Diffusion and STRING v8.2 were of 0.926 (0.727) for mental retardation, and 0.930 (0.476) for lymphoma, which are still good results. For five diseases, namely amyotrophic lateral sclerosis, Ehlers-Danlos syndrome, spastic paraplegia, Usher syndrome and Zellweger syndrome, the heterogeneous association network approach was actually able to rank all the seed genes in the first position of the ranking. On the other hand, the PageRank diffusion in the PPPIN achieved AUC scores above 0.9 only for two diseases: colorectal cancer with 0.912 and xeroderma pigmentosum with 0.98. The lowest AUC and MAP scores were obtained for amyotrophic lateral sclerosis (0.53 and 0.028) and spastic paraplegia (0.49 and 0.083). The PPPIN strategy could not rank any of the seed genes for amyotrophic lateral sclerosis in the top 10 positions and only one was identified in the first 20. Also, only one gene out of the 5 seeds for spastic paraplegia was ranked in the top 10/20. In this case, the performance for both diseases is comparable to the one obtained using the random seed sets (data now shown).

Confidence weights, number of iterations and diffusion rate We assessed the contribution of STRING's weights expressing the degree of confidence in the associations between genes to the performance of the prioritization method by diffusing the initial preference vector using the filtered disease-specific seed sets on the network after setting all associations' weights to 1. Although the resulting AUC and MAP scores (0.957 and 0.662) were not substantially different from the ones obtained using the confidence weights (0.962 ans 0.711), they actually reflected in less 9 genes ranked in the top 10 (data not shown). Overall, the number of genes in the top 20 was the same, with slight variations per disease. From the five diseases achieving maximum performance in the differentially association weighted setting, only for Ehlers-Danlos syndrome, spastic paraplegia and Zellweger syndrome these results could be maintained.

In both random walk approaches, the best results were achieved using a limited number of iterations. STRING v8.2 provided consistent and stable performance when varying the number of diffusion steps. On the PPPIN, the best ranking was always obtained using two iterations. It would then stabilize for larger numbers of steps, although measuring considerably lower in the evaluation, since it was never able to rank more than 289 or 346 genes - out of 526 - in the top 10 and top 20, respectively.

Regarding the parameter controlling the rate of diffusion, the Heat Diffusion method delivered quite similar performance for the set of heat coefficients tried: in STRING v8.2, resulting in AUC scores ranging from 0.960 to 0.962 for each diffusion coefficient, considering equal number of iterations; in the PPPIN, AUC scores ranging between 0.859 and 0.862 with 2 iterations, $N = 2$, and between 0.766 and 0.771 using 5, 10, 20 and 100 iterations.These results indicate its robustness to variations in this parameter. For PageRank with priors, the impact of the back probability value was not neglegible. For the lowest back probabilities (0.01 and 0.05) the scores were unstable leading to considerable performance

variations, even using STRING v8.2. For $\beta = \{0.1, 0.3, 0.5, 0.7, 0.9\}$, the PageRank AUC scores in STRING v8.2 varied between 0.936 and 0.961 considering the results obtained using the same number of iterations. In the PPPIN, PageRank obtained AUC scores between 0.859 and 0.861 using 2 iterations and ranging between 0.758 and 0.775 using 5, 10, 20 and 100 iterations.

4 Conclusions

Prioritization results confirmed our hypothesis that networks integrating gene associations retrieved or predicted using data from heterogeneous sources should be in general more informative and potentially able to perform better in the identification of genes associated with a given disease when compared to networks containing only physical interactions. Advantages of the former are supported by three key observations: (1) associations derived from the combination of several types of evidence should be more reliable and accurate; (2) heterogeneous data integration enables a better coverage of the genome and larger network density, confering robustness to noise; (3) confidence weights can be devised in order to differentiate associations and mitigate the impact of false positive associations, particularly when based on a limited number of sources.

Nevertheless, our analysis shows that heterogeneous association networks do not present sufficient guarantee for maximum performance by themselves. In fact, the network-based score measuring the degree of relatedness of each candidate gene with a given disease based on a set of known disease-related genes proved to play a major role. Essentially, based on the results we could conclude that in comparison to neighborhood-limited scores a network-based measure able to capture global connectivity properties by considering indirect associations between genes is not only (1) more robust, as it compensates for the sparsity related to direct associations and tackles the "small world" effect issue; but also (2) more informative, deriving a score based on a systemic view of the interactome. This claim has also been previously hinted at in [15, 16].

Propagation schemes tested in the computation of global network-based scores diffused an initial preference vector expressing the distribution of the known disease-related genes through the network using random walks. These methods compute fast using iterative procedures, even for large networks. Furthermore, we could verify that in the context of prioritization in association or physical interaction networks the maximum performance can be achieved using only a limited number of iterations. Heat Diffusion and PageRank with priors delivered high quality results and achieved similar performance under appropriate parameter settings, supporting the claim of equivalence [8, 25] for other approaches of the same kind, namely HITS with priors and K-Step Markov. The importance of confidence weights was inconclusive, as the difference in performance exhibited by our experiments was residual. We believe, however, that appropriate association confidence weights may improve accuracy of network-based prioritization results.

Acknowledgments This work was partially supported by FCT (INESC-ID multiannual funding) through the PIDDAC Program funds. JPG is the recipient of a doctoral grant supported by FCT (SFRH/BD/36586/2007).

References

1. NCBI Entrez Gene FTP Repository (Jan 2010), ftp://ftp.ncbi.nih.gov/gene/
2. Aerts, S., Van Loo, P., De Smet, F., Lambrechts, D., Maity, S., Tranchevent, L.C., De Moor, B., Coessens, B., Marynen, P., Hassan, B., Carmeliet, P., Moreau, Y.: Gene prioritization through genomic data fusion. Nature Biotechnology 24(5), 537–44 (2006)
3. Ashburner, M., Ball, C., Blake, J., Botstein, D., Butler, H., Cherry, J., Davis, A., Dolinski, K., Dwight, S., Eppig, J., Others: Gene Ontology: tool for the unification of biology. Nature Genetics 25(1), 2529 (2000)
4. Bader, G.D., Betel, D., Hogue, C.W.V.: BIND: the Biomolecular Interaction Network Database. Nucleic Acids Research 31(1), 248–250 (2003)
5. Breitkreutz, B.J., Stark, C., Reguly, T., Boucher, L., Breitkreutz, A., Livstone, M., Oughtred, R., Lackner, D.H., Bahler, J., Wood, V., Dolinski, K., Tyers, M.: The BioGRID Interaction Database: 2008 update. Nucleic Acids Research 36(suppl_1), D637–640 (2008)
6. Can, T., Çamoğlu, O., Singh, A.K.: Analysis of protein-protein interaction networks using random walks. In: Proceedings of the 5th International Workshop on Bioinformatics - BIOKDD '05. p. 61. ACM Press, New York, New York, USA (2005)
7. Chatr-Aryamontri, A., Zanzoni, A., Ceol, A., Cesareni, G.: Searching the protein interaction space through the MINT Database. Methods in Molecular Biology 484, 305–317 (2008)
8. Chen, J., Aronow, B.J., Jegga, A.G.: Disease candidate gene identification and prioritization using protein interaction networks. BMC Bioinformatics 10, 73 (2009)
9. Chung, F., Yau, S.: Coverings, heat kernels and spanning trees. Electronic Journal of Combinatorics 6, R12 (1999)
10. Francisco, A.P., Gonçalves, J.P., Madeira, S.C., Oliveira, A.L.: Using personalized ranking to unravel relevant regulations in the Saccharomyces cerevisiae regulatory network. In: Jornadas de Bioinformática 2009. Lisbon, Portugal (2009)
11. Franke, L., Bakel, H., Fokkens, L., Jong, D., E.d, Egmont-petersen, M., Wijmenga, C.: Reconstruction of a functional human gene network, with an application for prioritizing positional candidate genes. The American Journal of Human Genetics 78, 1011–1025 (2006)
12. Jensen, L.J., Kuhn, M., Stark, M., Chaffron, S., Creevey, C., Muller, J., Doerks, T., Julien, P., Roth, A., Simonovic, M., Bork, P., von Mering, C.: STRING 8–a global view on proteins and their functional interactions in 630 organisms. Nucleic acids research 37(Database issue), D412–6 (2009)
13. Kanehisa, M., Goto, S., Kawashima, S., Okuno, Y., Hattori, M.: The KEGG resource for deciphering the genome. Nucleic Acids Research 32(suppl_1), D277–280 (2004)
14. Kerrien, S., Alam-Faruque, Y., Aranda, B., Bancarz, I., Bridge, A., Derow, C., Dimmer, E., Feuermann, M., Friedrichsen, A., Huntley, R., Kohler, C., Khadake, J., Leroy, C., Liban, A., Lieftink, C., Montecchi-Palazzi, L., Orchard, S., Risse, J., Robbe, K., Roechert, B., Thorneycroft, D., Zhang, Y., Apweiler, R., Hermjakob, H.:

IntAct–open source resource for molecular interaction data. Nucleic Acids Research 35(suppl_1), D561–565 (2007)
15. Köhler, S., Bauer, S., Horn, D., Robinson, P.: Walking the interactome for prioritization of candidate disease genes. The American Journal of Human Genetics 82(4), 949958 (2008)
16. Linghu, B., Snitkin, E.S., Hu, Z., Xia, Y., Delisi, C.: Genome-wide prioritization of disease genes and identification of disease-disease associations from an integrated human functional linkage network. Genome biology 10(9), R91 (2009)
17. McKusick, V.A.: Mendelian Inheritance in Man and Its Online Version, OMIM. The American Journal of Human Genetics 80(4), 588–604 (2007)
18. von Mering, C., Jensen, L.J., Kuhn, M., Chaffron, S., Doerks, T., Krüger, B., Snel, B., Bork, P.: STRING 7–recent developments in the integration and prediction of protein interactions. Nucleic Acids Research 35(Database issue), D358–62 (2007)
19. Mishra, G.R., Suresh, M., Kumaran, K., Kannabiran, N., Suresh, S., Bala, P., Shivakumar, K., Anuradha, N., Reddy, R., Raghavan, T.M., Menon, S., Hanumanthu, G., Gupta, M., Upendran, S., Gupta, S., Mahesh, M., Jacob, B., Mathew, P., Chatterjee, P., Arun, K.S., Sharma, S., Chandrika, K.N., Deshpande, N., Palvankar, K., Raghavnath, R., Krishnakanth, R., Karathia, H., Rekha, B., Nayak, R., Vishnupriya, G., Kumar, H.G.M., Nagini, M., Kumar, G.S.S., Jose, R., Deepthi, P., Mohan, S.S., Gandhi, T.K.B., Harsha, H.C., Deshpande, K.S., Sarker, M., Prasad, T.S.K., Pandey, A.: Human protein reference database–2006 update. Nucleic Acids Research 34(suppl_1), D411–414 (2006)
20. Nitsch, D., Tranchevent, L.C., Thienpont, B., Thorrez, L., Van Esch, H., Devriendt, K., Moreau, Y.: Network analysis of differential expression for the identification of disease-causing genes. PloS ONE 4(5), e5526 (2009)
21. Schaefer, C.F., Anthony, K., Krupa, S., Buchoff, J., Day, M., Hannay, T., Buetow, K.H.: PID: the Pathway Interaction Database. Nucleic Acids Research 37(suppl_1), D674–679 (2009)
22. Tiffin, N., Andrade-Navarro, M.A., Perez-Iratxeta, C.: Linking genes to diseases: it's all in the data. Genome Medicine 1(8), 77 (Jan 2009)
23. Vanunu, O., Magger, O., Ruppin, E., Shlomi, T., Sharan, R.: Associating Genes and Protein Complexes with Disease via Network Propagation. PLoS Computational Biology 6(1) (2010)
24. Vastrik, I., D'Eustachio, P., Schmidt, E., Joshi-Tope, G., Gopinath, G., Croft, D., de Bono, B., Gillespie, M., Jassal, B., Lewis, S., Matthews, L., Wu, G., Birney, E., Stein, L.: Reactome: a knowledge base of biologic pathways and processes. Genome Biology 8(3), R39 (2007)
25. White, S., Smyth, P.: Algorithms for estimating relative importance in networks. In: KDD '03: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 266–275. ACM, New York, NY, USA (2003)
26. Wu, X., Jiang, R., Zhang, M.Q., Li, S.: Network-based global inference of human disease genes. Molecular Systems Biology 4(189), 189 (2008)
27. Xenarios, I., Rice, D.W., Salwinski, L., Baron, M.K., Marcotte, E.M., Eisenberg, D.: DIP: the Database of Interacting Proteins. Nucleic Acids Research 28(1), 289–291 (2000)
28. Yang, H., King, I., Lyu, M.: Diffusionrank: a possible penicillin for web spamming. In: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. p. 438. ACM (2007)

Collaboration-based Social Tag Prediction in the Graph of Annotated Web Pages

Hossein Rahmani[1], Behrooz Nobakht[1], and Hendrik Blockeel[1,2]

[1] Leiden Institute of Advanced Computer Science, Universiteit Leiden,
Niels Bohrweg 1, 2333 CA Leiden, The Netherlands
hrahmani@liacs.nl,bnobakht@liacs.nl,blockeel@liacs.nl
[2] Department of Computer Science, Katholieke Universiteit Leuven,
Celestijnenlaan 200A, 3001 Leuven, Belgium

Abstract. *Different approaches based on content or tag information have been proposed to address the problem of tag recommendation for a web page. In this paper, we analyze two approaches in a graph of web pages. Each node is a web page and edges represent hyperlinks. The first approach uses the content while the second one uses tag information in the graph. The second approach makes two assumptions about the tag set of two interacting nodes. The Tag Similarity Assumption claims that two interacting nodes discuss about rather similar topics; therefore, the chance of having more similar tag set is higher. The Tag Collaboration Assumption says that two interacting nodes complement each others topics. We apply algorithms such as Self Organizing Map (SOM), Reinforcement Learning (RL) and K-means clustering to compare methods on several datasets. We conclude that tag-based tag predictors outperform their content-based peers by more than ten percent with respect to the cosine similarity between predicted and actual tag sets.*

Key words: Social Tag Prediction, Tag Similarity Assumption, Tag Collaboration Assumption

1 Introduction

Social tagging systems have become increasingly popular in recent years. Various domain applications such as search engines, recommendation systems, spam detection and many others have improved their performance by considering these types of information.

Although using user's meta data has been shown useful for different purposes, these new types of describing resources have their own problems. The uncontrolled vocabulary nature of these data has inherent ambiguity and may make it hard to get a coherent vision of different users.

One possible way to solve the problems is to help users choose better informative tags. The system could suggest some "good tags" to users and users may or may not use those tags. In this paper, we proposed two different approaches for the task of social tag prediction in a graph of web pages. For each approach

there are different implementations. The first approach uses only the content of the web pages for the task of tag prediction. The second approach considers two assumptions about the tag set of two interacting web pages in a graph. The first assumption says that two interacting web pages have similar tag sets (Tag Similarity Assumption) while the second one assumes that two interacting web pages have "collaborative" tag sets (Tag Collaboration Assumption), collaborative meaning in this case that the tag sets somehow complement each other.

The rest of the paper is organized as follows: Section 2 briefly reviews motivations and related works. Section 3 discusses our approach for social tag prediction. In Section 4, first, we tune the parameters of our methods and then, we present experimental results on five web page graphs. We present our conclusions in Section 5.

2 Related Works

Various works have explored social annotations for different purposes. A lot of methods designed to improve web search results by considering data from social bookmarking systems ([1, 2]). Social annotation could be seen as a new way of organizing information and categorizing resources ([3, 4]). Users of social tagging systems could be connected to each other based on their areas of interests. The term Folksonomy, a combination of folk and taxonomy, was first proposed by [5]. A general introduction of folksonomy could be found in [6]. Ranking and recommender systems for folksonomies are proposed in [7, 8]. While the above mentioned approaches might look similar to our work but they are actually orthogonal, since the authors do not directly address the problem of predicting tags for resources.

The approaches in [9, 10] predicts tags based on the content and tag co-occurance respectively, while our methods consider also the neighborhood context of each web page for the tag prediction task. [11] proposes neighborhood-based tag prediction by using content similarity. They apply a straightforward scoring model to select the candidate tags, however, we use machine learning methods for tag prediction. [12] predicts tags from the set of 100 most frequent tags found in del.icio.us by training the binary classifier for each tag. We do not restrict ourselves to a predefined set of tags which are predicted.

Another important difference between our method and [11, 12] is that our method predicts the set of tags without considering any number of known tags for a given web page, whereas the methods in [11] and [12] are more "Social Tag Expansion" methods; they start with some known tags for a web page and then try to expand that set. In other words, we do not assume any knowledge about the document's own tags, in contrast to [11] and [12].

We introduce a new assumption "Tag Collaboration", which to our knowledge is not discussed in any previous work. We also define the "Topic Locality" characteristic of a web page graph, which is shown to affect the optimal parame-

ter settings and the performance of the approach. These novelties sets aside our methods from the related works.

3 Proposed Methods

The graph of web pages is represented by $G(P, E)$ in which P is a set of web pages and E is a set of interactions between web pages. Each $e_{pq} \in E$ shows a hyperlink between two web pages $p \in P$ and $q \in P$. Let T be the set of all the tags that occur in one dataset. Each classified web page $p \in P$ is annotated with a $|T|$-dimensional vector TS_p that indicates the tag set of this web page: $TS_p(t_i)$ is 1 if $t_i \in T$ is assigned to the web page p, and 0 otherwise. TS_p can also be seen as the set of all tags t_i for which $TS_p(t_i) = 1$. Similarly, the $|T|$-dimensional vector NB_p describes how often each tag occurs in the neighborhood (all the web pages that are reachable from p with a path of length at most 1) of web page p. $NB_p(t_i) = n$ means that among all the web pages that interact with p, n are annotated with tag t_i.

In this section, we discuss different approaches for solving the problem of social tag prediction in a graph of web pages. We analyze two main approaches for this purpose. The first approach uses the content while the second one uses tag information in the graph.

3.1 Content-Based Tag Predictor

In this approach, we use only the content of a web page for predicting its tags. We use the standard bag-of-words approach: after stemming and stop word removal, a vector is constructed in which each component is the frequency with which a particular word occurs on the page.

Most similar: Our first implementation for this approach is a standard nearest neighbor approach, which we refer to as "Most similar". In this implementation, first, we compare the content of the unannotated web page with the content of all the annotated web pages in the graph, using cosine similarity (Formula 1). After finding the most similar annotated web page, we select the top tags of the annotated web page and assign them to the unannotated one.

$$Cosine\ similarity\ (p, q) = \frac{p.q}{|p| * |q|} \tag{1}$$

K-means: In this approach, we first cluster the web pages in the graph based on their content. Then, we find the most frequently occurring tags in each cluster. The popular (most frequent) tags of each cluster will be assigned to all the unannotated web pages in that cluster.

We use K-means [13] for clustering the graph of web pages. We start clustering the network with k random centers and iteratively assign the web pages to

these k clusters based on the content similarity between web pages and the cluster centers. In each iteration, cluster centers move towards more balanced points in the cluster. As similarity metrics we use both Jaccard similarity (Formula 2) and Cosine similarity (Formula 1) of the tag sets.

$$Jaccard\ similarity\ (p,q) = \frac{|p \cap q|}{|p \cup q|} \tag{2}$$

3.2 Tag-Based Tag Predictors

Contrary to the previous approach, here, we use only tag information available in the graph for the task of tag prediction. We consider two assumptions behind the web page interactions. In the first assumption, two interacting web pages discuss about same topics and as a result their tag set will be similar (Tag Similarity assumption). Second assumption claims that interacting web pages complement the topics of each other and they do not necessarily have the same set of tags (Tag Collaboration assumption). We propose "Majority Rule Tag Predictor" as one of the possible implementations for *tag similarity* assumption. Two methods based on Reinforcement Learning (RL) and Self Organizing Map (SOM) are proposed for implementing *tag collaboration* assumption. We describe each proposed implementation in detail in the following:

Majority Rule Tag Predictor: This method simply implements tag similarity assumption by finding the most common tag(s) among the neighbors of the unannotated web page. Typically, a fixed number of tags is predicted, for instance the five most frequently occurring tags in the neighborhood are predicted for the unannotated web page.

The hypothesis behind *tag similarity* based approaches is that web pages with similar tags are always topologically close in the graph which all web pages in actual graphs not necessarily corroborate this hypothesis.

Reinforcement Based Tag Predictor: In this method, we try to quantify how strongly two tags t_i and t_j collaborate, in the following way. Let $TagColVal(t_i, t_j)$ denote the strength of collaboration between t_i and t_j.

We consider each annotated web page $p \in P$ in turn. If tag t_j occurs in the neighborhood of web page p (i.e., $NB_p(t_j) > 0$) then we increase the collaboration value between tag t_j and all the tags in TS_p:

$$\forall t_i \in TS_p : TagColVal(t_i, t_j) += \frac{NB_p(t_j) * R}{support(t_j)}$$

If tag t_j does not occur in the neighborhood of p ($NB_p(t_j) = 0$), we decrease the collaboration value between tag t_j and all the tags belonging to TS_p:

$$\forall t_i \in TS_p : TagColVal(t_i, t_j) -= \frac{P}{support(t_j)}$$

$support(t_j)$ is the total number of times that tag t_j appears on a side of an edge e_{pq} in the graph. R and P are "Reward" and "Punish" coefficients determined by the user.

Next, we determine the candidate tags for an unannotated web page p and rank them based on how well they collaborate with the neighborhood of web page p. As an example of a candidate tag strategy, consider Majority Rule: this method nominates all tags that appear in the direct neighborhood of the unannotated web page (and among these, will select the most frequently occurring ones). Here, we consider a number of extensions to this candidate tag strategy:

- First Tag Level Strategy (First-TL): This strategy selects the tags that appear in the direct neighborhood of the web page as candidate tags. This strategy nominates tags similar to the Majority Rule method.
- Second, Third, Fourth Tag Level Strategy (Second-TL, Third-TL, Fourth-TL): Define the n-neighborhood of a web page p as all the web pages that are reachable from p with a path of length at most n (thus, the 2-neighborhood includes all neighbors of neighbors of p, etc.). In Second-TL, Third-TL, Fourth-TL, all the tags occurring in the 2-, 3- or 4-neighborhood of p, respectively, are considered as candidate tags.
- All Tag Strategy (All-TL): All the tags are taken into account in this strategy.

After selecting candidate tags, we rank them based on how well they collaborate with the neighborhood of unannotated web page p. Formula (3) assigns a collaboration score to each candidate tag t_c:

$$Score(t_c) = \sum_{\forall t_j \in T} NB_p(t_j) * TagColVal(t_j, t_c) \tag{3}$$

High score candidate tag(s) collaborate(s) better with the neighborhood of p and are predicted as its tags.

We call the above method the "Reinforcement based tag predictor", as it is based on reinforcing collaboration values between tags as they are observed.

SOM Based Tag Predictor: Our second method makes use of a Self Organizing Map (SOM) for the task of tag prediction in a graph of web pages. We map the web page graph to a SOM as follows:

- Input Layer: The number of input neurons equals the number of tags in the web page graph. So, if $inputNeurons$ is the set of all neurons in the input layer then $|inputNeurons| = |T|$. The values we put in the input layer are extracted from the neighborhood tag vector of the web page: if $inputNeuron(i)$ is the i'th neuron in the input layer then $inputNeuron(i) = NB_p(t_i)$.
- Output Layer: The number of output neurons equals to the number of tags in the web page graph ($|outputNeurons| = |T|$). The values we put in the output layer are extracted from the tag vector of the web page: if $outputNeuron(i)$ is the i'th neuron in the output layer then $outputNeuron(i) = TS_p(t_i)$.

– Network Initialization: Weights of the neurons can be initialized to small random values; in our implementation we initialized all the weights to zero.
– Adaption: Weights of winner neurons and neurons close to them in the SOM lattice should be adjusted towards the input vector. The magnitude of the change decreases with time and with the distance from the winner neuron. Here, we take some new parameters into consideration which are $LearningRate(LR)$, $DecreasingLearningRate(DecLR)$ and $TerminateCriteria(TC)$ parameters. LR is the change rate of the weights toward the input vector and $DecLR$ determines the change rate of LR in different iterations. TC is the criteria in which the learning phase of SOM will terminate. Here, we think of TC as the minimum amount of change required in one iteration: when there is less change, the training procedure stops. We use Formula (4) for updating weights of output neurons.

$$W_{ij,New} = W_{i,j,Current} + LR * (NB_p(j) - W_{i,j,Current}) \tag{4}$$

– Testing: For each web page p in the graph that we did not use in the training phase, we find the Euclidean distance between NB_p and the weight vectors. We select the output neurons which have the shortest Euclidean distance to NB_p and predict them as the tag set of web page p. The number of predicted tags is fixed and determined by the user.

4 Empirical Results

4.1 Dataset

We were interested in graphs of web pages in which *a*) nodes are reasonably interconnected to each other at least as form of a tree; and *b*) nodes are tagged in a tagging system such as Delicious. Seemingly, there is no current data set available providing such information for web pages. Starting with ECML PKDD 09 Data Set [3] as the base, the first issue was to update the *weighted* tag assignments on web pages; we used URL's in the original data set to fetch their weighted tag assignments from Delicious [4]. The next consideration was the fact that there are many web pages in the data set that are *sparsely tagged* at Delicious; thus, we constrained the data set to the web pages annotated in Delicious with a minimum tag assignment weight. To build the desired graphs, starting from a web page in the data set, we included those neighboring URLs of the web page (and the links to them) that were tagged at Delicious. The same procedure was then applied to those neighboring URLs, and so on, up to a maximum crawling depth. Table 1 shows brief statistical information of datasets we used for evaluating our methods. Detailed information on data construction and preprocessing phase is available at `http://www.liacs.nl/~bnobakht/social-tag-prediction/`.

[3] `http://www.kde.cs.uni-kassel.de/ws/dc09/dataset/#files`
[4] `http://delicious.com/help/feeds`

Number of web page graph	27
Average page count of each web page graph	140.40
Average link count of each web page graph	311.03
Average number of tags for each page	6.92

Table 1. Statistical information of datasets.

4.2 Parameter Tuning

The methods we want to evaluate have some parameters for which good values need to be found. In order to tune the method's parameters, we select five different graphs and tune the parameters on those graphs and then use the tuned values for the other graphs.

While Majority Rule assigns only tags from the direct neighborhood to a web page, we are interested to find out whether using candidate tags from a wider neighborhood (including neighbors of neighbors) would be advantageous. We tested this by extending the Majority Rule so that it can consider not only direct neighbors, but also neighbors in the 2- or 3-neighborhood (as defined in Section 3.2). Figure (1) shows the effect of considering a wider neighborhood in Majority Rule in different datasets. There is no improvement over the base MR method by considering the second and third neighborhood level.

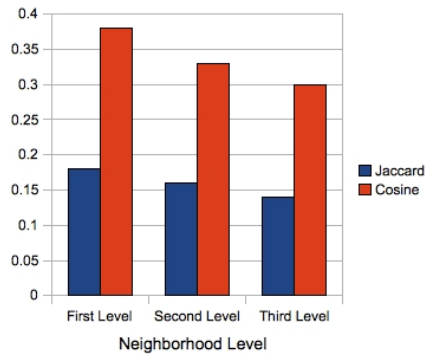


Fig. 1. Tuning "Neighborhood Tag Level" in the Majority Rule Tag Predictor method. There is no improvement over the base MR method by considering the second and third neighborhood level.

Figure 2 tunes the "Candidate Tag Strategy" parameter of RL method. The best value for this parameter is "First Level".

"Number of clusters (k)" is the parameter which should be tuned in the K-means algorithm. Before we tune this parameter, we introduce a *topic locality*

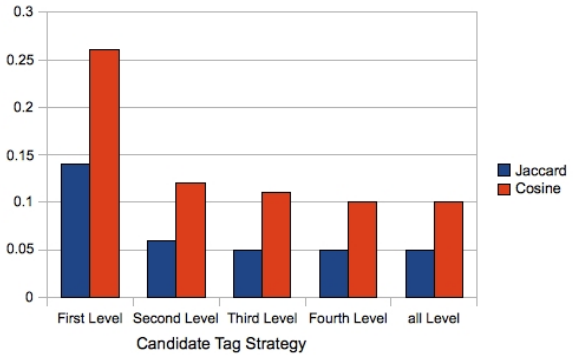


Fig. 2. Tuning "Candidate Tag Strategy" in the RL method. "First Level" tag strategy produces the best result.

feature in a graph of web pages. We define $topicLocality(G)$ as the number of distinct cluster tags divided by the number of clusters in the graph G (Formula 5).

$$TopicLocality(G) = \frac{Number\ of\ distinct\ cluster\ tags}{Number\ of\ clusters} \tag{5}$$

For instance, if all the web pages in the graph talk about the same topic then independent of different values for k, applying the K-means tag predictor will always lead to the same result. In this case we have small topic locality. But, if different parts of the graph discuss different topics, then we expect that by increasing the number of clusters, we increase the topic locality and this results in better tag prediction. Figure 3 shows the value of *topicLocality* in three different graphs. We cluster each graph into four clusters. In graph $G1$, all four clusters are about topic "Music", so the value of *topicLocality* is $\frac{1}{4}$ in this graph. Two clusters of graph $G2$ are about "Music", while the topics of the other two clusters are "Sport" and "Food". There are 3 distinct topics and 4 clusters, therefore the *topicLocality* of graph $G2$ equals $\frac{3}{4}$. In $G3$, all four clusters are about distinct topics, so the *topicLocality* of $G3$ equals $\frac{4}{4}$. So, choosing the right value for k in K-means algorithm is completely dependent to the *topic locality* of that specific graph.

Figure 4-(a) shows the result of applying K-means tag predictor with different values of k to graph $G(142, 292)$ with high *topic locality*. In a case of $k \leq 5$, there are k big "topic divergent" clusters, so the average cosine similarity is small (around 2%). As we increase the number of clusters the average cosine similarity value improves which means *topic locality* value of this graph is high. The best setting for K-means tag predictor is $60 \leq K \leq 64$.

Figure 4-(b) shows the effect of choosing different values for k in graph $G(362, 934)$ with low *topic locality*. In this graph, the topic of the most of the

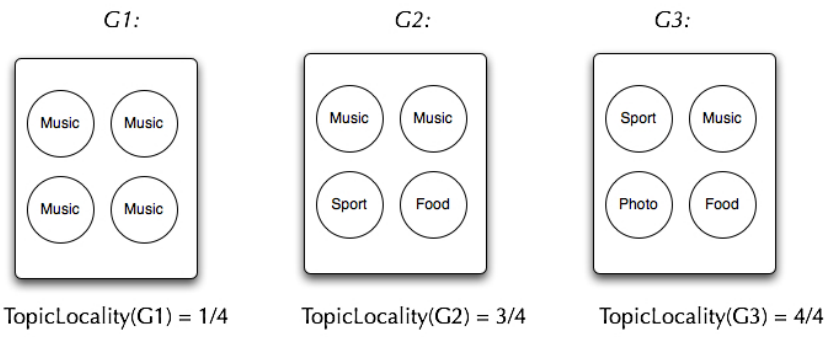


Fig. 3. *TopicLocality* character of different graphs.

web pages are similar to each other and changing the value of k does not effect the average cosine similarity a lot.

4.3 Comparison of Different Methods

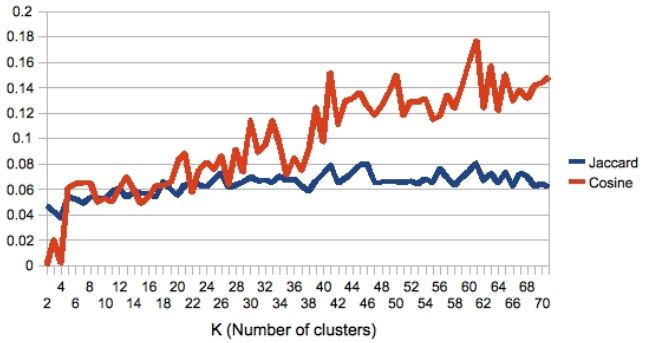
In this section, we compare content-based tag predictors (i.e., Most Similar and K-means) with tag-based tag predictors (i.e., Majority Rule, RL and SOM) on several datasets. We use average cosine similarity as the evaluation criterion. We predict 5 tags for each unannotated web page in all methods and then we compare the cosine similarity of different methods.

In the proposed implementations, we use the parameter values tuned in the previous section. Majority Rule (MR) selects the five most frequently occurring tags in the neighborhood of the web page in the network. As we discussed in the previous section, choosing the right value for k in K-means algorithm is completely dependent to the topic locality nature of that specific graph. In this section, we use fixed value for $k(= \frac{N}{5})$ for clustering all the graph. N equals the number of web pages in the graph.

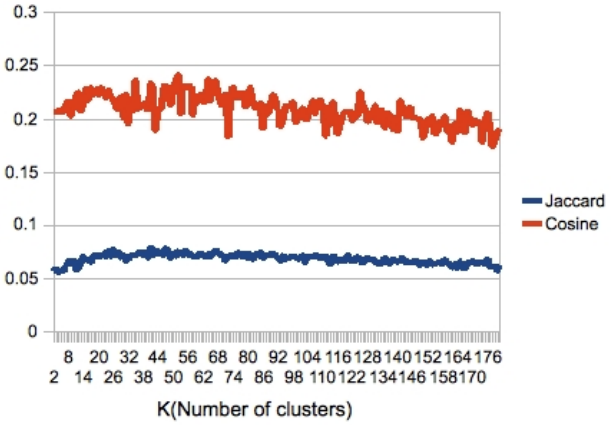
Figure 5 compares content-based tag predictors with tag-based tag predictors. Tag-based tag predictors outperform their content-based peers by more than 10 percent with respect to cosine similarity metric.

By assuming number of known tags for a given web page ("Social Tag Expansion" methods [11, 12]) or limit our methods predicting just small set of frequent tags (like [12]), we could expect a gain in precision; however this could drastically restrict the generality of our framework.

After analyzing each graph individually, we believe that there is no fixed approach (or parameters) which works best in all different types of web-page graphs. For example, considering "Topic locality" as a one out of many graph characteristics, in the graph with low Topic Locality (G1 in Figure 3) it might be better to use methods working based on the Tag Similarity assumption, while in high Topic Locality graphs (G3 in Figure 3), it is recommended to use Tag Collaboration based methods.



(a) Graph $G(142, 292)$ with high "Topic Locality" characteristics.



(b) Graph $G(362, 934)$ with low "Topic Locality" characteristics.

Fig. 4. Tuning "Number of Clusters" in the K-means algorithm.

5 Conclusion

To our knowledge, this is the first study that considers graph of annotated web pages for the task of tag prediction. We proposed two different approaches for the task of social tag prediction in a graph of web pages. For each approach, we recommend different implementations. The first approach uses only the content of the web pages for the task of tag prediction. For this approach we propose "Most Similar" and "K-means" methods. Contrary to the first approach, the second approach uses only tag information available in the graph for the task of tag prediction. It considers two assumptions about the tag set of two interacting web pages in a graph. The first assumption says that two interacting web pages have similar tag sets (Tag Similarity Assumption) while the second one assumes

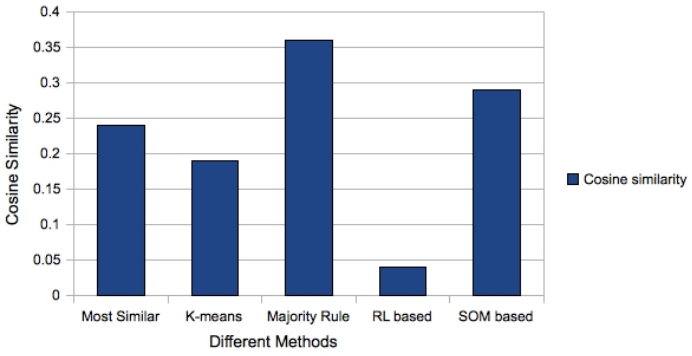


Fig. 5. Compare content-based tag predictors (i.e.,Most Similar and K-means) with tag-based tag predictors (i.e., Majority Rule, RL and SOM) on the different datasets. Tag-based tag predictors outperform their content-based peers more than ten percent with respect to cosine similarity.

that two interacting web pages have collaborative tag sets (Tag Collaboration Assumption). We proposed "Majority Rule" method as one of the possible implementations for that similarity assumption. This method simply predicts most frequently occurring tags in the neighborhood of unannotated web page. We used two machine learning methods Self Organizing Map (SOM) and Reinforcement Based Tag Predictor for implementing tag collaboration assumption. Both methods first learn the collaboration value between each pair of tags and then at prediction time, they rank candidate tags based on how well they collaborate with the neighborhood of unannotated web page.

We compared content-based tag predictors with tag-based tag predictors and we found out that Tag-based tag predictors outperform their content-based peers by more than 10 percent with respect to cosine similarity metric. Among the tag-based tag predictors, Majority Rule method predicts the best tags for unannotated web pages which means Tag Similarity Assumption dominates Tag Collaboration Assumption in the graph of web pages. So, in general, web pages in the our dataset tend to discuss more about similar topics rather than complementary topics.

We also analyzed each graph individually and we concluded that graph characteristics have direct impact on choosing the right method for social tag prediction. We observed that in low Topic Locality graphs, results of Tag Similarity methods outperform the results Tag Collaboration methods while in graphs with high topic locality, it is better to apply Tag Collaboration methods.

Funding

This research is funded by the Dutch Science Foundation (NWO) through VIDI grant 639.022.605.

References

1. Han, P., Wang, Z., Li, Z., Kramer, B., Yang, F.: Substitution or complement: An empirical analysis on the impact of collaborative tagging on web search. In: WI '06: Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence, Washington, DC, USA, IEEE Computer Society (2006) 757–760
2. Bao, S., Xue, G., Wu, X., Yu, Y., Fei, B., Su, Z.: Optimizing web search using social annotations. In: WWW '07: Proceedings of the 16th international conference on World Wide Web, New York, NY, USA, ACM (2007) 501–510
3. Aliakbary, S., Abolhassani, H., Rahmani, H., Nobakht, B.: Web page classification using social tags. In: CSE '09: Proceedings of the 2009 International Conference on Computational Science and Engineering, Washington, DC, USA, IEEE Computer Society (2009) 588–593
4. Zubiaga, A., Martínez, R., Fresno, V.: Getting the most out of social annotations for web page classification. In: DocEng '09: Proceedings of the 9th ACM symposium on Document engineering, New York, NY, USA, ACM (2009) 74–83
5. Vander Wal, T.: Folksonomy definition and wikipedia. (November 2005)
6. Quintarelli, E.: Folksonomies: power to the people. ISKO Italy-UniMIB Meeting (2005)
7. Hotho, A., Jäschke, R., Schmitz, C., Stumme, G.: Information retrieval in folksonomies: Search and ranking. In Sure, Y., Domingue, J., eds.: ESWC. Volume 4011 of Lecture Notes in Computer Science., Springer (2006) 411–426
8. Jschke, R., Marinho, L.B., Hotho, A., Schmidt-Thieme, L., Stumme, G.: Tag recommendations in folksonomies. In Kok, J.N., Koronacki, J., de Mntaras, R.L., Matwin, S., Mladenic, D., Skowron, A., eds.: Knowledge Discovery in Databases: PKDD 2007, 11th European Conference on Principles and Practice of Knowledge Discovery in Databases, Warsaw, Poland, September 17-21, 2007, Proceedings. Volume 4702 of Lecture Notes in Computer Science., Springer (2007) 506–514
9. Chirita, P.A., Costache, S., Nejdl, W., Handschuh, S.: P-tag: large scale automatic generation of personalized annotation tags for the web. In: WWW '07: Proceedings of the 16th international conference on World Wide Web, New York, NY, USA, ACM (2007) 845–854
10. Sigurbjörnsson, B., van Zwol, R.: Flickr tag recommendation based on collective knowledge. In: WWW '08: Proceeding of the 17th international conference on World Wide Web, New York, NY, USA, ACM (2008) 327–336
11. Budura, A., Michel, S., Cudr-Mauroux, P., Aberer, K.: Neighborhood-based tag prediction. In Aroyo, L., Traverso, P., Ciravegna, F., Cimiano, P., Heath, T., Hyvnen, E., Mizoguchi, R., Oren, E., Sabou, M., Simperl, E.P.B., eds.: ESWC. Volume 5554 of Lecture Notes in Computer Science., Springer (2009) 608–622
12. Heymann, P., Ramage, D., Garcia-Molina, H.: Social tag prediction. In Myaeng, S.H., Oard, D.W., Sebastiani, F., Chua, T.S., Leong, M.K., eds.: SIGIR, ACM (2008) 531–538
13. MacQueen, J.B.: Some methods for classification and analysis of multivariate observations. In Cam, L.M.L., Neyman, J., eds.: Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability. Volume 1., University of California Press (1967) 281–297

How much linguistics do we need in order to understand online opinions?

Carlos Rodríguez

Research Center, Barcelona-Media, Spain

Abstract

The vast amount of online opinionated text has driven the interest of an active research community that exploits this user-generated content to gather market information and create business intelligence applications. State-of-the-art Natural Language Processing techniques can provide a level of text interpretation that might be adequate for certain tasks, but there is room for improvement over the current methods which are based on pre-existing knowledge, such as prior polarity lexicons and domain ontologies. The crucial question is how much resource-intensive linguistic processing is needed to understand what people are talking about, and how do they feel about it. A principled combination of symbolic and stochastic approaches that is guided by bootstrapping existing and extensive Web 2.0 resources seems to be a good compromise when full text interpretation is not available or practical.

Automatic Sentiment Monitoring of Specific Topics in the Blogosphere

Fernanda S. Pimenta, Darko Obradović, Rafael Schirru,
Stephan Baumann, and Andreas Dengel

German Research Center for Artificial Intelligence (DFKI),
Knowledge Management Department
& University of Kaiserslautern,
Computer Science Department
Kaiserslautern & Berlin, Germany
{fpimenta,obradovic,schirru,baumann,dengel}@dfki.uni-kl.de

Abstract. The classification of a text according to its sentiment is a task of raising relevance in many applications, including applications related to monitoring and tracking of the blogosphere. The blogosphere provides a rich source of information about products, personalities, technologies, etc. The identification of the sentiment expressed in articles is an important asset to a proper analysis of this user-generated data. In this paper we focus on the task of automatic determination of the polarity of blogs articles, i. e., the sentiment analysis of blogs. In order to identify whether a piece of text expresses a positive or negative opinion, an approach based on word spotting was used. Empirical results on different domains show that our approach performs well if compared to costly and domain-specific approaches. In addition to that, if we consider an aggregation of a set of documents and not the polarity of each individual document, we can achieve an accuracy distribution around 90% for specific topics of a certain domain.

Keywords: opinion mining, sentiment analysis, blogosphere

1 Introduction

In order to achieve a better analysis and organization of the large amount of online documents available nowadays, it is very useful to classify texts according to the sentiment that they express [1]. The sentiment analysis of texts can be applied to various tasks such as text summarization, management of online forums, and monitoring of the acceptance of a given product or brand through the tracking of discussions on weblogs [2]. The blogosphere provides a rich source of information about products, personalities, technologies, etc. The identification of the sentiment expressed in blogs is an important asset to a proper analysis of this user-generated data.

Not only big companies benefit from sentiment analysis, but also politicians, journalists, advertisers, and market researchers. The research in this field encompasses diverse domains such as movies (e. g., [1],[3]), cars, books, travel (e. g., [4]),

and many other products and services (e. g., [5]). The large amount of available information sources and different domains make an automatic approach for the sentiment analysis of the blogosphere indispensable. In this paper, we focus on the problem of classifying a text according to its polarity, which can be one out of positive, negative, or neutral, in a non-domain-specific and scalable way. Some known methods were implemented based on word spotting for the realization of this task and performed an evaluation of them using datasets from different domains.

The remainder of this article is structured as follows. In Section 2, we present related work in the field of sentiment analysis. We describe the methods implemented for the classification of text according to its polarity in Section 3. Next, in Section 4 we perform an evaluation of the methods. In Section 5 we consider how sentiment analysis can be used to monitor the blogosphere considering an aggregation of articles in topics. Then we present our findings and our ideas for future work in Section 6.

2 Related Work

Words and expressions that compose a text possess an evaluative character that varies not only in degree, but also in polarity [6]. A positive polarity means a positive evaluation and a negative polarity means a negative evaluation. In the sentiment analysis field, a large amount of work focuses on the classification of text according to its polarity. Identifying whether a text is either positive, negative, or neutral usually is done with word spotting techniques or machine learning. Word spotting techniques rely on sentiment bearing words and expressions that are either present in an affective lexicon or have their sentiment captured by an automatic approach.

Turney and Littman [6] proposed a method to automatically predict the polarity score of a word or phrase by its statistic association with a set of negative and positive paradigm words. This strategy is called Semantic Orientation from Association (SO-A) . The SO-A of a word/phrase is calculated by the difference between its power of association with the set of positive and its power of association with the negative set. They used two different measures to calculate the association: pointwise mutual information (PMI) and latent semantic analysis (LSA). With a different idea, Pang et al. [1] applied machine learning techniques to perform sentiment analysis in movie reviews. They employed Naïve Bayes, maximum entropy classification, and support vector machines, and although not as good as for topic categorization, the results were satisfactory. Gamon [5] also successfully used machine learning for the classification of consumer reviews, and besides predicting whether a review was positive or negative, it established a ranking (from 1 to 4) on it. The author manage to improve his SVM approach by also taking into account the effects of valence shifters over words and expressions.

Nigam and Hurst [7] presented a system to automatically detect polar expressions about a given topic through the integration of a shallow NLP polar

language extraction system and a machine learning based topic classifier. The results of their experiments show that if considered separately, the polarity classifier performs better than when applied together with the topic classifier. In the field of weblogs, Durant and Smith [8] applied a Naïve Bayes classifier together with a forward feature selection technique to identify the political sentiment of weblog posts. Their classifier performed well (even outperforming SVM), but their focus was a little bit different. They aimed to predict the left or right political alignment of posts. A very similar work to the one of Durant and Smith [8], but with the same task as ours (identifying positive and negative sentiment in blogs), is the one presented by Melville et al. [2]. They introduced a framework which uses background lexical information together with supervised learning as an approach to sentiment classification. Their results show that the approach is a good alternative to reducing the burden of labeling many examples in the target domain. However, like many other machine learning approaches, their experiments rely on well-balanced and structured datasets, many times from a unique domain or topic. Besides that, the previously mentioned studies take into account the polarity of individual documents, not of an aggregation of a set of documents, an approach that is considered in this paper.

3 Sentiment Classification

The sentiment analysis of a text can be performed based on the sentiment bearing terms (words or expressions) that comprise such text, e. g., using word spotting techniques. Through the counting of terms it is possible to classify the text according to its polarity. Counting positive and negative terms is a very simple technique proposed in [4] and [9] and may well be used to classify entire documents. Different from the approaches based on machine learning, term counting does not require training and it is suitable even when training data is not available. If the majority of the sentiment bearing terms of a text is positive, the text is considered positive. Otherwise, if the majority of these terms is negative, the text is classified as negative. If there is some kind of balance between positive and negative terms, the text is considered neutral. Term counting relies on words and expressions that are either present in an affective lexicon or have their polarity captured by an automatic approach. We implemented these two types of term counting approaches and called them *lexicon based approach* and semantic orientation from association approach.

3.1 Lexicon Based Approach

First of all, we perform sentence segmentation and part-of-speech tagging (POS tagging) over the text we want to classify using the JTextPro text processing toolkit [10]. Then, for each of the terms considered sentiment relevant in the text, we consult an affective lexicon that contains polarity information about these terms. We have chosen *SentiWordNet* [11] as our affective lexicon since it is a lexical resource freely available for educational and research purposes.

We use here *SentiWordNet* 1.0 (the latest version available at the time of our experiments). Through the combination of the results produced by eight ternary classifiers, *SentiWordNet* associates for each of the *synsets* of *WordNet* (version 2.0) three scores related to polarity properties (positive, negative, and objective) that each ranges from -1 to 1. For this approach, identifying the polarity score of a text consists then in calculating the average polarity score of the terms that comprise it. We considered here two variant methods depending on which terms should be used in the calculation. In the first, only adjectives and adverbs of the sentences are taken into account (we call it LB_AdjAdv). The second one is a modification of LB_AdjAdv (we call it LB_AdjAdvMod), in which the effect of contextual valence shifters on the polarities of the adjectives and adverbs are considered. The concept of contextual valence shifters was introduced in [3]. They consist of negations, intensifiers and diminishers and they flip, increase, or decrease the polarity score of a sentiment term. When either an adjective or an adverb is found, we look for contextual valence shifters that occur near it and, if found, the weights of the valence shifters are multiplied with the original score of the adjective/adverb. Table 1 shows an example of the impact of valence shifters on the word *cool*, which originally has the positive polarity score of 0.5 according to *SentiWordNet*.

Table 1. Example of the effect of valence shifters over the word *cool*.

Valence Shifter	Score
None	0.5
Negation (e. g., not)	-0.5
Intensifier(e. g., very)	1.0
Diminisher(e. g., slightly)	0.25

3.2 Semantic Orientation from Association Approach

Like in the above mentioned approach, we first segment the text and then apply POS-tagging on it using the JTextPro text processing toolkit [10]. Second, we use patterns of POS tags defined in [4] for extracting phrases from the processed text (Table 2). The JJ tags are adjectives, the NN tags are nouns, the RB tags are adverbs, and the VB tags are verbs [1]. For each phrase, we then calculate the SO-A of it using as the measure of association the Pointwise Mutual Information (SO-PMI). Based on [6], in order to calculate the PMI of each phrase, we issue queries to a search engine (in our case, Yahoo![2]) and count the number of hits the set of paradigm words gets alone and the number of hits it gets with the phrase. Let $Pwords$ be the set of paradigm positive words and $Nwords$ the set

[1] For a complete reference on the POS tags, see [12]

[2] Using the Yahoo! API available at http://developer.yahoo.com/

of paradigm negative words. The SO-PMI of a phrase, i. e., its polarity score, is defined as

$$SO - PMI(phrase) = \log_2 \frac{hits(phrase, Pwords)hits(Nwords)}{hits(phrase, Nwords)hits(Pwords)}$$

Table 2. Patterns of POS tags for extracting two-Word Phrases [4].

	First Word	Second Word	Third Word (not extracted)
1.	JJ	NN or NNS	anything
2.	RB, RBR, or RBS	JJ	not NN nor NNS
3.	JJ	JJ	not NN nor NNS
4.	NN or NNS	JJ	not NN nor NNS
5.	RB, RBR or RBS	VB, VBD, VBN, or VBG	anything

The polarity of the entire text is then calculated by the average of the SO-PMI scores of all the phrases that comprise it. We call this method SO_PMI.

4 Evaluation

In this section, we present experiments and analyses of the application of the implemented methods. We perform two sets of experiments. The first compares all the methods implemented and choose the best of them. The comparison of all methods is only performed with one data set because of time limitations to execute the SO_PMI method. The API used to issue queries to the Yahoo! search engine has a limit of 5000 queries a day, and to calculate the SO-PMI of all the phrases in all the data sets, would take a quite long time (around forty days).

4.1 Data sets

Our motivating application is to perform the sentiment analysis of blog posts. Blogs are much more diverse and complex in structure than reviews. However, since there is a great amount of sentiment annotated data sets regarding reviews and they have been used extensively in previous sentiment analysis works, we decided also to use these data sets in our empirical evaluation. We have used the following publicly available data sets.

Amazon Reviews The data set for the first set of experiments is comprised of 1000 Amazon camera and photo product reviews and it was first presented in [13]. Each review consists of a rating that ranges from 1 to 5 stars. Reviews with rating values greater than 3 were labeled as positive, those with rating values of

less than 3 were labeled negative, and the rest discarded because their polarity was considered ambiguous. We make this assumption about the ratings based on previous works that have already used this dataset (e. g., [13] and [14]), although it is well known that users rate items with different personal scales and this issue should be considered when estimating the relevance of items for a certain user [15]. In the end we have 500 positives and 500 negatives reviews for this dataset.

Convote This data set was introduced in [16] and consists of automatically transcribed political debates classified according to whether an utterance is in support of a motion, or in opposition to it. There were in total 701 utterance, 426 in support and 275 in opposition.

Movie Reviews Provided by [1], this data consists of 1000 positive and 1000 negative reviews from the Internet Movie Database. Positive labels were assigned to reviews that had a rating above 3.5 stars and negative labels were assigned to the rest. We use version 2.0 of this dataset in our experiments.

Service reviews This data set contains reviews of six different domains and was provided by Whitehead and Yaeger [17]. The domains, as well the amount of positive and negative reviews of each domain are summarized in Table 3

Table 3. Domains of the Whitehead and Yaeger [17] data set

Domain	Positive reviews	Negative reviews	Total
Camp	402	402	804
Doctor	739	739	1478
Drug	401	401	802
Lawyer	110	110	220
Radio	502	502	1004
Tv	235	235	470

4.2 Results

We carried out two sets of experiments, one with the Amazon reviews data set and the other with the remaining data sets. In the first set of experiments, we used the accuracy of the classification in order to determine which approach works best on the data set. We present in Table 4 the results of these experiments based on the accuracy of classifying the reviews correctly (as either positive or negative), i. e., the total number of reviews correctly classified against the total number of reviews.

Table 4. Comparing accuracy of different approaches to sentiment classification with the Amazon reviews data set.

Method	**Accuracy**
LB_AdjAdv	61%
LB_AdjAdvMod	63%
SO_PMI	51%

It can be seen in Table 1 that the accuracy for the LB_AdjAdvMod is the highest. Although there is no huge difference between LB_AdjAdv and LB_AdjAdvMod, the addition of contextual valence shifters improves the accuracy of classification, as already shown in [3]. The surprise here was the poor performance of the method using SO_PMI. Using SO-PMI, Turney and Littman [6] obtained in their experiments an accuracy around 80% to automatically predict the polarity score of words. In our experiments, the accuracy of this method is as good as a random classifier (that would achieve 50% of accuracy). This is probably due to the fact that the scores computed with SO-PMI are not always trustworthy. One possible problem is that the number of hits returned by a search engine is not known to be 100% reliable and hence the calculation of the SO-PMI of the phrases would not be 100% reliable too.

Since the LB_AdjAdvMod method was the best in the first set of experiments, we choose it to be used as the classifier for the second set of experiments. We performed the classification on the rest of the non-blog data sets and the results concerning accuracy are shown in Table 5.

Table 5. Accuracy of LB_AdjAdvMod approach in different data sets.

Data Set	**Accuracy**
Convote	55%
Movie Reviews	60%
Camp	66%
Doctor	72%
Drug	61%
Lawyer	74%
Radio	61%
Tv	63%

The results demonstrate that for all data sets, the classifier performs better than a random classifier (with a baseline of 50%). The algorithm achieves an accuracy of 74% with the Doctor data set which is satisfactory compared to the methods that exist so far. However, for the Convote and Movie Reviews data sets the results are still very close to the random classifier. The poor results for

the Convote data set may be related to the fact that it consists of transcribed spoken political debates and not originally written text. This could influence the performance of the classifier since it was created aiming at written language, not spoken. On the other hand, for the Movie Reviews data set, maybe the problem was the fact that sometimes a review contains negative words describing the plot of the movie, but this does not mean that the review is negative [3].

5 Sentiment Monitoring of Topics

The accuracy of sentiment analysis is still not satisfactory when compared with other automatic classifiers. Natural language is highly complex, the state of the art not reliable, and some critics doubt it will ever work since this task is difficult even for humans. However, it is possible to use sentiment analysis to monitor the distribution of polarity over a set of documents of a specific topic instead of individual documents. Our hypothesis is to consider the distribution of polarity over an aggregation of documents in order to achieve much more reliable results with today's mediocre classifier accuracies. In order to analyze this idea, we performed a new set of experiments with the LB_AdjAdvMod method using a data set comprised of blog articles from different topics of a given domain.

5.1 Android blogs data set

To test our best approach in the domain of blogs, we have annotated a set of blog articles with sentiment scores. The original blog data collection used here was presented in Schirru et al. [18] and comprises blog articles categorized into topics. Per topic we read each article and annotated it manually as either positive, negative, or neutral. Table 6 shows the topics that comprise the final labeled set.

Table 6. Topics of the Android blogs data set.

Topic	Number of Articles
cupcake	118
dev-phone-block	77
htc-magic	68
uk-app-market	54
amazon-deal	48
robot-control	48
windows-mobile	24
gartner-study	17
iverse-comics	14

5.2 Evaluation

We classified the articles in the Android blogs data set using the LB_AdjAdvMod method and compared the resulting classification with the manually created ground truth (GT). The distribution of polarity over the set of articles of each specific topic was used then as an initial evaluation. Considering an interval from -0.1 to 0.1 for the neutral class, we aggregated the articles according to their polarities in three classes: negative, neutral, and positive. The difference between the total number of articles in each GT class and the total number of articles in our classifier's class is calculated. Then, we calculate the penalty cost to equalize the LB_AdjAdvMod distribution with the GT distribution. Considering that the cost to transfer one article from neutral to any of the other classes (or vice-versa) is 0.5, and from positive to negative (or vice-versa) is 1, the accuracy distribution of our classifier will be the total penalty cost divided by the total number of articles of the topic. As a baseline, we take the classification of a random classifier (RC) that distributes evenly the articles among the three polarity classes. For this classifier, the worst case is when all the articles in the GT belong to the positive class (or the negative class). However, even in the worst case, the accuracy distribution of the RC will never be lower than 0.5. In Table 7, we have the distributions for the topic *dev-phone-block*. This topic

Table 7. Distribution of the *dev-phone-block* articles into polarity classes according to three classifiers

Classifier	**Negative**	**Neutral**	**Positive**
GT	56	20	1
LB_AdjAdvMod	42	26	9
RC	25.67	25.67	25.67

concerns the announcement of the android market blocking some new merchant applications which caused the frustration of many developers. As we can see by the distributions of the LB_AdjAdvMod and the GT, the classifier captures well the tendency of the overall sentiment towards the topic (mostly negative in this case). Calculating the accuracy distribution for the LB_AdjAdvMod method, we get 85.71% against 64.29% for the RC, showing that our method performs better than chance.

Table 8 shows the values for the accuracy distribution for the classification of the LB_AdjAdvMod method and the RC considering the GT of the Android blogs data set. For most of the topics our classification performs well, however, for a few of them it is as good as RC. Reading the articles from topics like *uk-app-market* and *amazon-deal* we can observe that there is no tendency to a more positive or more negative sentiment towards the topic. These articles are more objective and don't have good indications of sentiment.

Table 8. Accuracy distribution per topic of the LB_AdjAdvMod method and the RC for the Android blogs data set.

Topic	LB_AdjAdvMod	Random Classifier
cupcake	87.71%	66.81%
dev-phone-block	85.71%	64.29%
htc-magic	92.65%	83.33%
uk-app-market	70.37%	70.37%
amazon-deal	85.42%	86.46%
robot-control	80.21%	71.88%
windows-mobile	85.42%	81.25%
gartner-study	97.06%	74.51%
iverse-comics	89.29%	82.14%

6 Conclusion and Future Work

We have implemented methods for sentiment analysis using word spotting approaches. Empirical results on different domains show that although our best approach performs well if compared to costly and domain-specific approaches, it is still not satisfactory. However, if we consider the distribution of polarity over an agreggation of documents we have much more reliable results than considering the classification of each document separately. We analyzed this distribution in a set of articles of different topics of a certain domain and we noticed that our method can provide good indications for the sentiment monitoring of the blogosphere. We believe this method is also useful in domains where the number of positive and negative samples is not normally balanced (e. g., the movies domain).

Increasing the list of contextual valence shifters and using an affective lexicon with higher coverage are possible ways of improving our method. In our experiments, we used as neutral threshold the value 0.1 (i. e., the article with a score between -0.1 and 0.1 belongs to the neutral class). It would be interesting to perform tests to find out what value for the neutral threshold would result in better accuracy. Another good direction for future work is to take into account only terms near the keywords related to an article's topic to calculate the polarity of the article.

References

1. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up? Sentiment classification using machine learning techniques. In: Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP). (2002) 79–86
2. Melville, P., Gryc, W., Lawrence, R.D.: Sentiment analysis of blogs by combining lexical knowledge with text classification. In: KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, New York, NY, USA, ACM (2009) 1275–1284

3. Kennedy, A., Inkpen, D.: Sentiment classification of movie reviews using contextual valence shifters. Computational Intelligence **22** (2006) 2006
4. Turney, P.D.: Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In: ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, Morristown, NJ, USA, Association for Computational Linguistics (2002) 417–424
5. Gamon, M.: Sentiment classification on customer feedback data: noisy data, large feature vectors, and the role of linguistic analysis. In: In COLING. (2005) 841–847
6. Turney, P., Littman, M.: Measuring praise and criticism: Inference of semantic orientation from association. ACM Transactions on Information Systems **21** (2003) 315–346
7. Nigam, K., Hurst, M.: Towards a robust metric of polarity. In Shanahan, J.G., Qu, Y., Wiebe, J., eds.: Computing Attitude and Affect in Text: Theory and Applications. Volume 20 of The Information Retrieval Series. Springer-Verlag, Berlin/Heidelberg (2006) 265–279
8. Durant, K.T., Smith, M.D.: Predicting the political sentiment of web log posts using supervised machine learning techniques coupled with feature selection. In Nasraoui, O., Spiliopoulou, M., Srivastava, J., Mobasher, B., Masand, B.M., eds.: WEBKDD. Volume 4811 of Lecture Notes in Computer Science., Springer (2006) 187–206
9. Turney, P.D., Littman, M.L.: Unsupervised learning of semantic orientation from a hundred-billion-word corpus. CoRR **cs.LG/0212012** (2002)
10. Phan, X.H.: Jtextpro: A java-based text processing toolkit (2006) http://jtextpro.sourceforge.net/.
11. Esuli, A., Sebastiani, F.: SentiWordNet: A publicly available lexical resource for opinion mining. In: Proceedings of LREC-06, 5th Conference on Language Resources and Evaluation, Genova, IT (2006) 417–422
12. Taylor, A., Marcus, M., Santorini, B.: The penn treebank: An overview (2003)
13. Blitzer, J., Dredze, M., Pereira, F.: Biographies, bollywood, boomboxes and blenders: Domain adaptation for sentiment classification. In: In ACL. (2007) 187–205
14. Blitzer, J., Crammer, K., Kulesza, A., Pereira, O., Wortman, J.: Learning bounds for domain adaptation. In: In Advances in Neural Information Processing Systems. (2008)
15. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Trans. Knowl. Data Eng. **17**(6) (2005) 734–749
16. Thomas, M., Pang, B., Lee, L.: Get out the vote: Determining support or opposition from Congressional floor-debate transcripts. In: Proceedings of EMNLP. (2006) 327–335
17. Whitehead, M., Yaeger, L.: Building a general purpose cross-domain sentiment mining model. Computer Science and Information Engineering, World Congress on **4** (2009) 472–476
18. Schirru, R., Obradović, D., Baumann, S., Wortmann, P.: Domain-specific identification of topics and trends in the blogoshere. To appear in: Perner, P. (ed.) ICDM 2010. LNCS (LNAI), vol. 6171, Springer, Heidelberg (2010)

Different Aggregation Strategies for Generically Contextualized Sentiment Lexicons

Stefan Gindl

Department of New Media Technology, MODUL University Vienna, Austria

Abstract. Sentiment detection has gained relevance in the last years due to the vast amount of publicly available opinion in the form of Web forums or blogs. Yet, it still suffers from the ambiguity of language, lowering the efficacy and accuracy of sentiment detection systems. Thus, it is important to also invoke context information to refine the initial values of sentiment terms. Moreover, domain-independence is desirable to avoid using a topic determination beforehand. This work investigates strategies for extracting non-generic features to be integrated into a so-called contextualized sentiment lexicon, capable of getting the context correctly and assigning sentiment terms the proper sentiment value. The proposed approach will be applied in an online-media aggregation and visualization portal, covering a vast number of news media sources.

1 Introduction

Sentiment detection handles affect expressed in written text, more exactly it tries to classify documents into positively, negatively or neutrally opinionated. The classification can either be coarse-grained (i.e. positive, negative, neutral) or fine-grained (i.e. strong-positive, weak-positive, etc.). The research area experienced a leap in relevance with the upcoming availability of online opinions in reviews, forums or blogs. Applications range from the political area (tracking a political campaign online) over the economic area (acceptance studies for new products or services) to the purely scientific application, helping to understand human language. Thus, sentiment detection can play a major role in Web mining systems. It also adds value to Social Web applications. Trend analyses on fast moving platforms such as www.twitter.com become possible; websites hosting images or videos (such as www.flickr.com or www.youtube.com) can be exploited to measure the affect of the community towards celebrities or popular technical devices.

Many approaches rely on so-called sentiment lexicons, containing terms assumed to express sentiment. Sentiment lexicons suffer from term ambiguity - one and the same term can have different meanings under different circumstances. Table 1 shows three sentence, where one and the same sentiment term can be used in positive and negative context. The intuitively negative term “repair” can be used positively, when a person is satisfied with his/her repaired car. “Unpredictable” applied to the movie genre refers to an exciting movie; on the other

hand, if the breaks of a car are unpredictable, this is normally something undesirable. Finally, the term "peace" will be express a positive fact in the most cases. Yet, it can also refer to a negative state, such as in the sentence "This peace is a lie".

Positive	Negative
The **repair** of my car was satisfying.	I had many complaints after my camera's **repair**.
This movie's plot is **unpredictable**.	The breaks of this car are **unpredictable**.
The long **peace** brought wealth and safety to the people.	This **peace** is a lie.

Table 1. Examples for sentiment terms occurring in positive and negative contexts.

This work examines possible refinement strategies of the already existing context-sensitive sentiment detection system described in [7]. It takes into account the context of a sentiment term, and, based on the context, refines the sentiment value of the term. Naïve Bayes as a simple, fast and yet powerful technique serves as the method to train the model. To overcome the effects of domain-specificity the approach also merges features of the trained models and creates a domain-independent model. In the presented paper refinement strategies for creating a domain-independent lexicon are discussed, together with a preliminary evaluation of the planned strategies.

Temporal Sentiment Analysis Applied to Online Media

The proposed system will be used for temporal sentiment detection in the so-called "Media Watch on Climate Change". This portal aggregates climate change related issues and provides efficient visualization means, such as a semantic map for related keywords with strong media coverage and an ontology map for relations among significant phrases.

The sentiment map in the upper left corner of Figure 1 allows for tracing the sentiment towards relevant topics. For example, the phrases "oil spill" and "gulf oil" receive clearly negative media attention, whereas the term "Hayward" received positive attention until May 10, which turns into negative afterwards. Such a tool, i.e. accurate sentiment detection combined with efficient visualization techniques, strongly supports research on relevant topics and offers a specialized view on the online world.

During the U.S. elections 2008 another portal website using a former version of the proposed appraoch traced media attention towards the presidential candidates. Figure 2 shows the main window of the portal, with the presidential candidates in the upper part, a list of used media sources in the middle and the

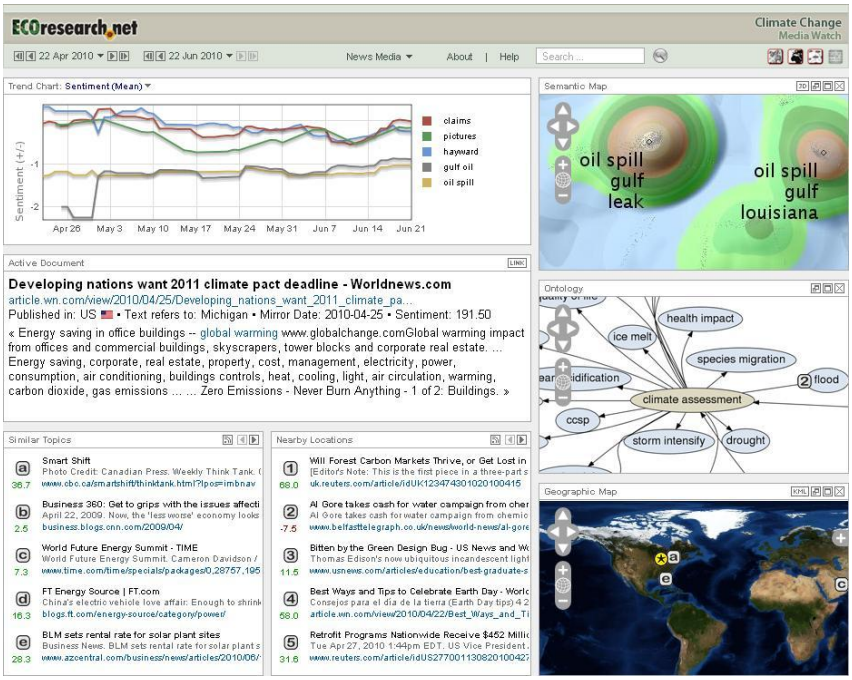


Fig. 1. The Media Watch on Climate Change, www.ecoresearch.net/climate/; see the sentiment map

sentiment map at the bottom. Such tools can complement or even replace traditional opinion surveys, and are a permanent source of feedback during a political campaign. Adapted to different application fields they can support enterpises to trace their reputation (e.g. in connection with the current oil spill in the Gulf of Mexico) or to measure the acceptance of a previously launched new product in the online community.

The paper is structured as follows: Section 2 summarizes existing work, Section 3 outlines the already existing approach and the refinement strategies. The evaluation follows in Section 4. Section 5 concludes the paper and contains an outlook on further work regarding the discussed refinement strategies.

2 Related Work

Sentiment detection as a research area dates back to the 1990s with the work of Wiebe [20] and Hatzivassiloglou and McKeown [9]. In [20] Wiebe started to identify subjective sentences, whereas Hatzivassiloglou and McKeown exploited syntactical relations to identify sentimental adjectives [9]. Turney and Littman

apply two different association measurements to identify new sentimental terms in [17]. In [13] Pang and Lee present a fine-grained approach to detect the exact sentiment (i.e. the star rating) of reviews using Support Vector Machines. Subrahmanian and Reforgiato base sentiment detection on a syntactical level by using adjective-verb-adjective combinations [16].

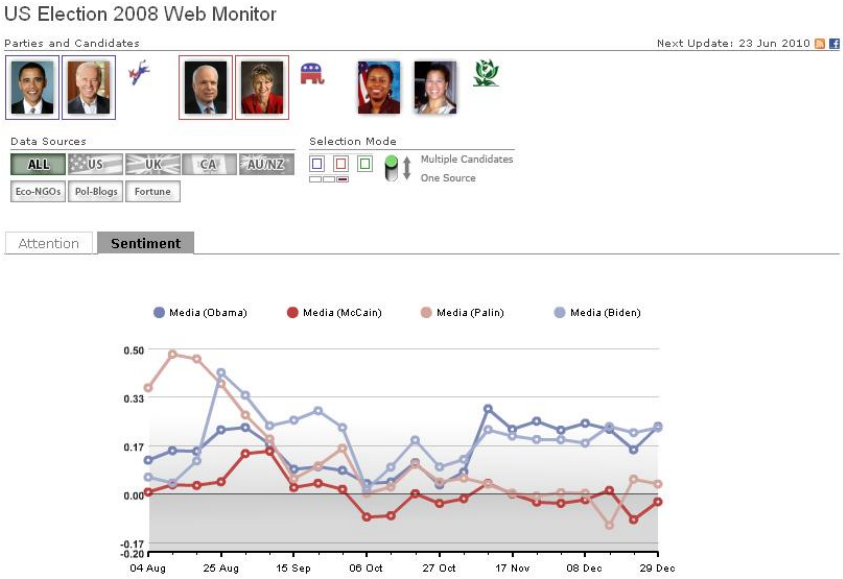


Fig. 2. The US Election 2008 Web Monitor, www.ecoresearch.net/election2008/; see the sentiment map

Some works also use context information to refine sentiment indicators. According to Nasukawa and Yi [12] sentiment detection is a three step process, where the identification of sentiment expressions is followed by the determination of their polarity and strength. The last step of the procedure identifies the subject the sentiment terms are related to. They model such relationships for verbs, which either directly transfer their own sentiment or another term's sentiment to the subject. With this model they are capable of treating expressions such as t_i *prevents trouble* [12]. The verb *prevents* passes the opposite sentiment of the term *trouble* to the target t_i. Sentence particles different from verbs directly transfer their sentiment to the subject. Kim and Hove [10] specify subjects with a Named-Entity-Recognition and assign them the overall sentiment value of the sentence. A list of 44 verbs and 34 adjectives expanded by WordNet [6] synonyms and antonyms serves as sentiment lexicon. To handle complex sentence structures such as "the California Supreme Court *disagreed* that the state's

new term-limit law was *unconstitutional*" [10] they developed a strategy, where several negative sentiment terms in one and the same sentence eliminate each other. Polanyi and Zaenen present a number of "contextual valence shifters" in their eponymous work [14]. Agarwal et al. propose syntactical capturing of context in [1]. Wilson et al. evaluate a large number of textual features, including context, in [21] on different machine learning algorithms; they use a two-stage process, firstly filtering neutral expressions from polar ones and afterwards disambiguating the sentiment of the polar expressions. In [22] they present a similar procedure with an expanded set of machine learners.

Turney and Littman [17] use Pointwise Mutual Information (PMI) and Latent Semantic Analysis (LSA) to identify sentiment terms in a large Web corpus. Terms with sufficient co-occurrence frequency with one of 14 paradigm terms (i.e. a gold standard list of seven positive and negative terms) are assigned the same sentiment value as the respective paradigm term. Evaluated on the General Inquirer [15] PMI shows results comparable with the algorithm of Hatzivassiloglou and McKeown [9]. Using three different extraction corpora and the sentiment lexicon of [9] Turney and Littman show that PMI does not outperform Hatzivassiloglou's and McKeown's algorithm but is more scalable [19]. LSA also provided better results, but was not as scalable as PMI too. In [18] Turney uses the same techniques to identify new sentiment terms from a paradigm list of only two terms (*excellent* and *poor*). This procedure performed well on the review corpus. Beineke et al. re-interpret the previously discussed mutual association as a Naïve Bayes approach [2]; they also expand this perspective (which is an unsupervised approach) and create a supervised approach using labeled data.

Lau et al. [11] prove the importance of context by applying three different language models, whereof one is an inferential language model sensible for context. According to their evaluation the inferential language model outperforms the other two models, emphasizing the importance of context. Bikel and Sorensen apply a simple feature selection together with a perceptron classifier to reviews from Amazon.com [3]. They use all tokens with an occurrence frequency higher than four and achieve an accuracy of 89% in their experiments. Denecke [4] applies a machine learning approach to multi-lingual sentiment detection using movie reviews from six different languages. Google Translator (www.google.com/language_tools) translates foreign-language documents into English. The feature selection procedure extracts a total of 77 features out of four superclasses [4]: (1) the frequency of word classes (i.e. the number of verbs, nouns, etc.), (2) polarity scores for the 20 most frequent words and the averages scores for all verbs, nouns and adjectives are calculated using SentiWordNet [5]; other features are (3) the frequency of positive and negative words according to the General Inquirer and (4) textual features such as the number of question marks. Using all features the Simple Logistic classifier of the WEKA tool[8] reaches exorbitantly good results when applied to native English documents. When applied to non-native, translated documents the results are still higher than the baseline demonstrating the efficacy of using a lexical resource such as SentiWordNet.

Our contextualization method is different from the presented context-aware approaches. For example, we do not use linguistic relations such as synonymy as Esuli and Sebastiani in [5]. Furthermore, we also do not transfer sentiment from sentiment terms to subjects as done in [12], nor do we filter polar from neutral expressions as or use predefined syntactical features [21, 22]. Instead, the proposed method considers the term's context based on discriminators identified in the text and adjusts its sentiment value accordingly.

3 Methodology

The work is based on [7] and can be roughly divided into three steps (also see Figure 3). The first step comprises the enrichment of an initial sentiment lexicon with contextual information. The initial lexicon is a lexicon based on sentimental terms from the General Inquirer [15]. We applied "reverse lemmatization" on these terms, which adds inflected forms to the initial terms. The second step is the application of the created contextualized sentiment lexicon on unknown documents, using the Naïve Bayes technique to recalculate the original sentiment values in the sentiment lexicon. The last step comprises the identification of context features applicable across the domains of the training corpora. This step results in the creation of a generic contextualized lexicon. We compare the improvement achieved with this approach using a lexical algorithm as our baseline. This algorithm sums up the sentiment values of all sentiment terms occurring in a document:

$$Sent(doc) = \sum_{i=1}^{n} Sent(t_i)$$

$$Sent(t_i) = \begin{cases} 1, \text{ if } t_i \text{ is a positive term} \\ -1, \text{ if } t_i \text{ is a negative term} \\ 0, \text{ if the term is neutral} \end{cases}$$

In case of a negation trigger preceding a sentiment term its value is multiplied by -1. In the following, we describe each of these steps in more detail:

Generation of the contextualized lexicon The system identifies ambiguous terms in the initial sentiment lexicon by analyzing their usage in a labeled training set. The training set consists of documents with positive and negative labels. A sentiment term with equally high frequency in both parts is considered to be an ambiguous term. All ambiguous terms identified with that process undergo a so-called "contextualization". This means, that the system identifies terms frequently co-occurring with the ambiguous term in positive/negative reviews (i.e. context terms). The contextualization creates a contextualized lexicon. This lexicon stores the probability that a certain ambiguous term in combination with certain context terms is normally used in positive/negative reviews.

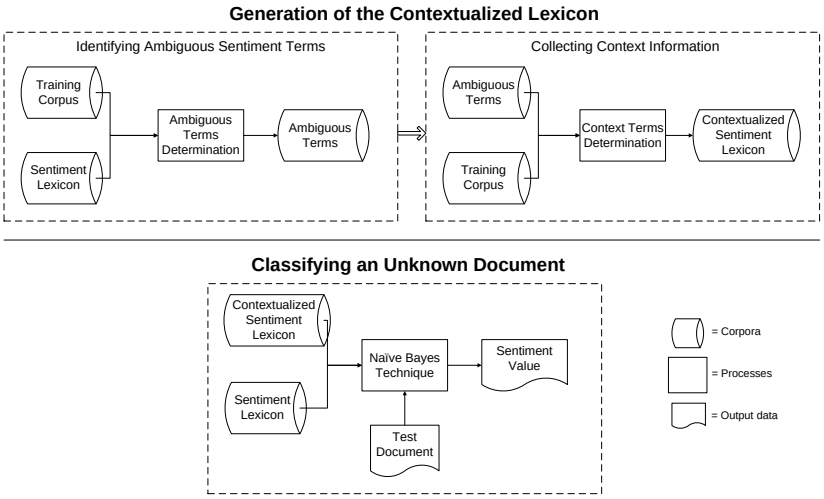


Fig. 3. Creation and application of a contextualized sentiment lexicon.

Application on unknown documents Each time a sentiment term occurs in a new document, the contextualized sentiment lexicon is consulted and decides, if the term is ambiguous. For non-ambiguous terms the lexicon returns the original sentiment value of the term. In case of an ambiguous term the system analyzes the context of the document. It uses the ten strongest context sentiment terms and calculates the probability of the ambiguous term being positive/negative given these ten context terms.

The system calculates an ambiguous term's sentiment given context **c** using the Naïve Bayes formula (c_i is a single context term):

$$p(Sent^+|\mathbf{c}) = \frac{p(Sent^+) \cdot \prod_{i=1}^{n} p(c_i|Sent^+)}{\prod_{i=1}^{n} p(c_i)}$$

The resulting value is the final sentiment value of the ambiguous term. Finally, the sentiment values of all sentiment terms (ambiguous and non-ambiguous) are summed up. The sum is the overall sentiment of the document.

Figure 4 shows an example of the context-sensitive sentiment detection. The system analyzes the document and finds the sentiment term "repair", which turns out to be ambiguous. So, it also analyzes the context, i.e. all other terms of the document. It identifies the three context terms "friendly", "quickly", and "reliable" as indicators for a positive meaning of "repair". Thus, the system assigns it a positive sentiment value and classifies the whole document as being positive. Note that the example is very simple - in reality a document usually contains more sentiment terms, both ambiguous and non-ambiguous.

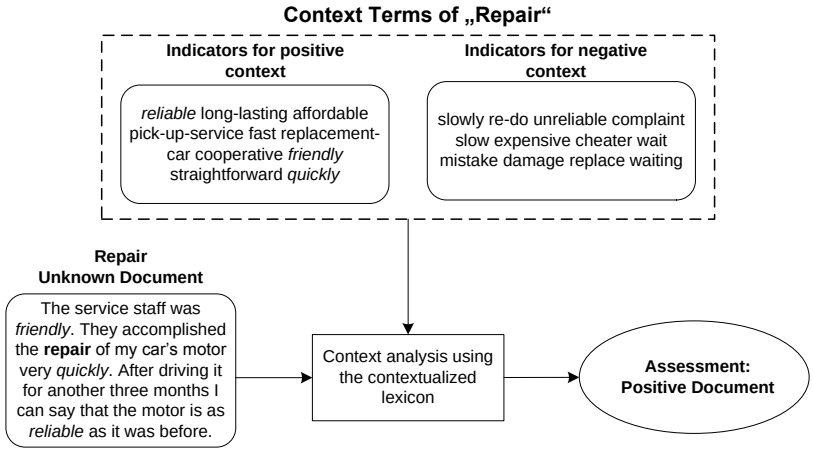


Fig. 4. Context invocation for the ambiguous term **repair** in an unknown document.

Identifying Generic Features Generic features are context terms which can be used across domains. Having obtained the contextualized lexicons from several training corpora the system distinguishes between three types of context term categories:

- **Helpful**: Using a helpful sentiment term improves the efficacy of sentiment detection.
- **Neutral**: These terms do not change the efficacy.
- **Harmful**: Harmful terms reduce the efficacy.

The categorization into helpful, neutral and harmful is accomplished as follows: if a review has been classified incorrectly by our baseline (i.e. the lexical algorithm explained at the beginning of this section), but correctly by the Naïve Bayes approach, the context terms of all ambiguous terms in this document are considered as helpful terms. If it has been correctly classified by the baseline but is incorrectly classified by Naïve Bayes all context terms are considered as harmful. Neutral context terms are those occurring in documents where Naïve Bayes and the baseline deliver the same classification. Using such a procedure means that a term helpful in document A can be neutral or even harmful in document B. A special exclusion strategy decides which of the harmful terms should be discared, and thus also their occurrences as helpful or neutral terms.

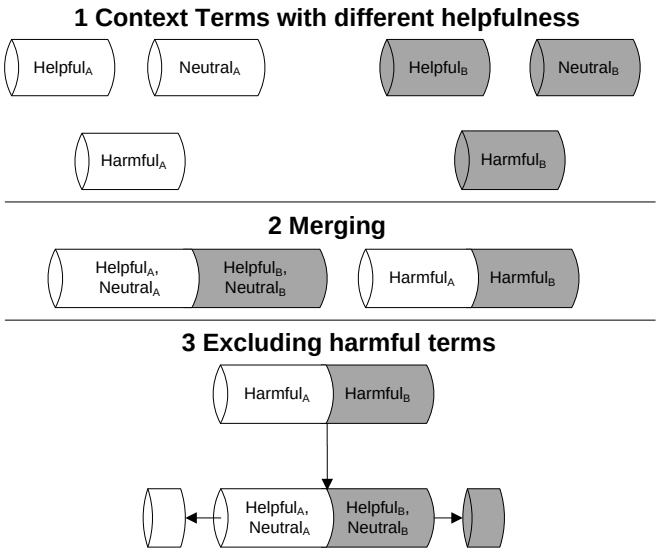


Fig. 5. Filtering harmful terms

4 Evaluation

We evaluated the contextualization refinements on the same corpora as in [7], which are a set of 2 500 products reviews from Amazon[1] and 1 800 holiday reviews from TripAdvisor[2] (which we call the "Amazon" and the "TripAdvisor" corpus later on). We accomplished a 10-fold cross-validation on both evaluation sets. A simple lexical approach serves as the baseline for the evaluation, summing up sentiment values of the sentiment terms occurring in the document to be classified. The sentiment values come from the initial lexicon described in Section 3.

We tested the following strategies for the exclusion of harmful terms:

- C_{all}: no harmful terms are excluded.
- $C \setminus H$: even terms with a single harmful occurrence are excluded.
- $C = \{c | \frac{F(c|\neg h)}{F(c|h)} > 5\}$: if a term has been helpful/neutral, but also has a harmful occurrence, its frequency in helpful/neutral cases must be five times higher than in harmful cases.
- $C = \{c | \frac{F(c|\neg h)}{F(c|h)} > 10\}$: if a term has been helpful/neutral, but also has a harmful occurrence, its frequency in helpful/neutral cases must be ten times higher than in harmful cases.

[1] amazon.com
[2] tripadvisor.com

– H: only terms with harmful occurrences are used.

In Table 2 we give the results (i.e. the F-measures) for all tested exclusion strategies. For each corpus we distinguish between positive and negative and list the F-measure for each type (indicated by $\oplus$ and $\ominus$). The evaluation shows that excluding harmful terms requires great care. Removing all terms with harmful occurrences ($C \setminus H$) gives worse results than leaving them untouched (C_{all}). Setting the ratio of non-harmful terms to harmful terms to high (i.e. > 10) gives the same results as keeping all harmful terms. Using only terms having harmful occurrences lowers the evaluation results strongly. Yet, the results are not low enough to judge them as completely useless. Finally, using a weaker ratio (i.e. > 5) delivers the best results.

		C_{all}	$C \setminus H$	$C = \{c \mid \frac{F(c\mid\neg h)}{F(c\mid h)} > 5\}$	$C = \{c \mid \frac{F(c\mid\neg h}{F(c\mid h} \} > 10$	H
Amazon	$\oplus$	0.68	0.68	**0.69**	0.68	0.58
	$\ominus$	0.74	0.73	**0.75**	0.74	0.72
TripAdvisor	$\oplus$	**0.84**	**0.84**	**0.84**	**0.84**	0.81
	$\ominus$	0.78	0.78	**0.79**	0.78	0.78

Table 2. F-Measures achieved with different exclusion strategies

5 Conclusion & Further Work

The evaluation showed that particular aggregation strategies improve the overall result for sentiment detection using contextualized lexicons. Their sole impact is not too large, but they should be regarded as an integral component of a battery of refinement strategies for generically contextualized sentiment detection.

Future work comprises the investigation on further, more potential aggregation strategies. Moreover, an investigation of the semantic and syntactical sentence structure will be accomplished. The idea is that certain sentence types might mislead sentiment detection. For example, sentences which are too short or too long, or are in another way distorted might be counterproductive for sentiment detection. If used anyways those sentences worsen classification results. Sentiment detection would benefit from a-priori filtering of these. Machine-learning methods can accomplish this task.

References

1. Apoorv Agarwal, Fadi Biadsy, and Kathleen R. Mckeown. Contextual phrase-level polarity analysis using lexical affect scoring and syntactic N-grams. In *EACL '09: Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 24–32, Morristown, NJ, USA, 2009. Association for Computational Linguistics.

2. Philip Beineke, Trevor Hastie, and Shivakumar Vaithyanathan. The sentimental factor: Improving review classification via human-provided information. In *ACL '04: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 263, Morristown, NJ, USA, 2004. Association for Computational Linguistics.
3. Daniel M. Bikel and Jeffrey Sorensen. If we want your opinion. In *ICSC 2007. International Conference on Semantic Computing*, pages 493–500, Irvine, CA, September 2007.
4. Kerstin Denecke. How to assess customer opinions beyond language barriers? In *Third International Conference on Digital Information Management*, pages 430–435. IEEE, November 2008.
5. Andrea Esuli and Fabrizio Sebastiani. SentiWordNet: A publicly available lexical resource for opinion mining. In *Proceedings of the 5th International Conference on Language Resources and Evaluation*, pages 417–422, 2006.
6. C. Fellbaum. WordNet - An electronic lexical database. *Computational Linguistics*, 25(2):292–296, 1998.
7. Stefan Gindl, Albert Weichselbraun, and Arno Scharl. Cross-domain contextualization of sentiment lexicons. In *ECAI 2010: Proceedings of the 19th European Conference on Artificial Intelligence*, in press.
8. Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The WEKA data mining software: An update. *SIGKDD Explorations*, 11(1):10–18, 2009.
9. Vasileios Hatzivassiloglou and Kathleen R. McKeown. Predicting the semantic orientation of adjectives. In *Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics*, pages 174–181, Morristown, NJ, USA, 1997. Association for Computational Linguistics.
10. Soo-Min Kim and Eduard Hovy. Determining the sentiment of opinions. In *COLING '04: Proceedings of the 20th international conference on Computational Linguistics*, page 1367, Morristown, NJ, USA, 2004. Association for Computational Linguistics.
11. R.Y.K. Lau, C.L. Lai, and Yuefeng Li. Leveraging the web context for context-sensitive opinion mining. In *Computer Science and Information Technology, 2009. ICCSIT 2009. 2nd IEEE International Conference on*, pages 467–471, Aug. 2009.
12. Tetsuya Nasukawa and Jeonghee Yi. Sentiment analysis: Capturing favorability using natural language processing. In *K-CAP '03: Proceedings of the 2nd international conference on Knowledge capture*, pages 70–77, New York, NY, USA, 2003. ACM.
13. Bo Pang and Lillian Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 115–124, Morristown, NJ, USA, 2005. Association for Computational Linguistics.
14. Livia Polanyi and Annie Zaenen. Contextual valence shifters. In *Computing Attitude and Affect in Text: Theory and Applications*, The Information Retrieval Series, 2006.
15. Philip J. Stone, Dexter C. Dunphy, and Marshall S. Smith. *The General Inquirer: A computer approach to content analysis*. M.I.T. Press, Cambridge, Massachusetts, 1966.
16. V.S. Subrahmanian and Diego Reforgiato. AVA: Adjective-Verb-Adverb combinations for sentiment analysis. *Intelligent Systems, IEEE*, 23(4):43–50, July-August 2008.

17. P.D. Turney and M.L. Littman. Unsupervised learning of semantic orientation from a hundred-billion-word corpus. Technical report, National Research Council, Institute for Information Technology, 2002.
18. Peter D. Turney. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In *ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 417–424, Morristown, NJ, USA, 2002. Association for Computational Linguistics.
19. Peter D. Turney and Michael L. Littman. Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems*, 21(4):315–346, 2003.
20. Janyce M. Wiebe. Tracking point of view in narrative. *Computational Linguistics*, 20(2):233–287, 1994.
21. Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 347–354, Morristown, NJ, USA, 2005. Association for Computational Linguistics.
22. Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. Recognizing contextual polarity: An exploration of features for phrase-level sentiment analysis. *Computational Linguistics*, 35(3):399–433, 2009.

Towards an Evaluation Framework for Topic Extraction Systems for Online Reputation Management*

Enrique Amigó[1], Damiano Spina[1], Bernardino Beotas[2], and Julio Gonzalo[1]

[1] Departamento de Lenguajes y Sistemas Informáticos
Universidad Nacional de Educación a Distancia
C/Juan de Rosal, 16
28020 Madrid, España
{enrique,damiano,julio}@lsi.uned.es
[2] Grupo ALMA
C/Valentín Beato, 23
28037 Madrid, España
b.beotas@almatech.es

Abstract. This work present a novel evaluation framework for topic extraction over user generated contents. The motivation of this work is the development of systems that monitor the evolution of opinionated topics around a certain entity (a person, company or product) in the Web. Currently, due to the effort that would be required to develop a gold standard, topic extraction systems are evaluated qualitatively over cases of study or by means of intrinsic evaluation metrics that can not be applied across heterogeneous systems. We propose evaluation metrics based on available document metadata (link structure and time stamps) which do not require manual annotation of the test corpus. Our preliminary experiments show that these metrics are sensitive to the number of iterations in LDA-based topic extraction algorithms, which is an indication of the consistency of the metrics.

* This work has been partially supported by Alma Technologies and the Spanish Government (projects Webopinion and Text-Mess/Ines)

1 Introduction

The growing interest on monitoring opinions in the Web 2.0 is well known. Online Reputation Management consists of monitoring the opinion of Web users on people, companies or products, and it is already a fundamental tool in corporate communication. A particularly relevant problem is to detect new topics or opinion trends which deserve the attention of communication experts, such as a burst of tweets or blog entries about a controversial issue about a company, or a defect of a product. A system that assists a communication expert should be able to detect (particularly new) topics, tag them in an interpretable way, cluster documents related to each topic and analyze the evolution of topics over time. What makes this a distinctive problem is the fact that documents are naturally multi-topic: relevance of a document for a topic may be even sub-sentential. This problem is sometimes referred to as Temporal Text Mining [1,2].

Models and systems to solve these tasks are recently starting to appear in scholar publications. But a major bottleneck so far is the absence of a benchmarking test suite to evaluate and compare systems. Creating such a gold standard is, in fact, a complex task: defining the set of topics in a document stream is a subtle task, because topics tend to co-occur in documents and the appropriate level of granularity in topic and sub-topic distinctions is something fuzzy to fix. For similar reasons it is also difficult, once the set of topics is established, to decide which documents talk about each of the topics and how central is each document to each of the topics that the document discusses. In the absence of a gold standard, extrinsic precision-recall based metrics can not be applied.

For this reason, current systems are evaluated informally via use cases, or otherwise using intrinsic evaluation measures which are specific to the model being tested.

There are, however, basic restrictions on how a good system should behave. For instance, documents which share outlinks to the same web pages should tend to be more related than documents which do not share outlinks. This type of information has not been yet used by current topic detection systems, because they relate together only a small subset of the documents. This information, however, might be used as a (limited) evaluation or validation mechanism to optimize system parameters. In this paper we address the task of defining an evaluation methodology based on this idea, and check its suitability on an LDA-based approach to topic detection over time.

2 State of the art

We will start with an overview of models to solve the task, and then we will summarize the evaluation methodologies used so far and discuss their limitations.

2.1 Topic detection models

The most basic approaches for topic monitoring focus on word frequency. The assumption is that frequent words indicate, in general, salient topics in a document

collection. Some available web services are Blogpulse Trends[3], Mood Views[4] and Blogscope[5]. Brooks and Montanez showed that frequent words (extracted according to tf.idf), produce tags that generate document clusters with more cohesion than user tags in blogs [3].

Gruhl included the temporal dimension in his model by extracting topic terms with frequency peaks over time [4]. Chi considered also the distribution of terms across blogs [5]. He assumed that topics gain prominence in blog subsets. His model consists of computing the singular values of the time-blog frequency matrix . Mei et al. combine topological information with the temporal dimension [6]. Their model employs the EM algorithm to identify the topic distributions along time and location that maximize the likelihood of word occurrences. An interesting feature of this model is that it assumes that several topics can appear in the same document.

Many novel proposals are currently based on the LDA (Latent Dirichlet Allocation) model [7]. As well as Mei's approach, LDA is a probabilistic model that estimates the distribution θ_d of topics for each document d and the distribution of words for each topic. The particularity of LDA is that the distribution parameters are generated by a Dirichlet distribution with certain hyperparameters that are stated a priori.

One example of these models is TOT (Topic Over Time) [8]. The most characteristic aspect of this work is that the temporal variable is added to the LDA model, assuming that topics follow a Beta distribution along time. One drawback in this work is that all the document collection must be processed for inferring temporal distribution when new documents appear in the input stream. The model *Dynamic Topic Model* [9] tries to solve it by estimating topic distributions for each time slot independently. After this, the model employs temporal series techniques in order to analyze topic evolution. Another model that tackle this issue is *On-line LDA* [10]. This model states that the knowledge produced over a time slot represents the a priori knowledge for the next time slot. This idea allows to process new documents without reprocessing the whole collection. However, an addition mechanism to detect new topics along the time becomes necessary. Another interesting model based on LDA is denominated *Multiscale Topic Tomography*[11]. In this approach the topic distribution includes different granularity levels.

2.2 Evaluation approaches

The main bottleneck in this area of research is the absence of a common evaluation methodology to compare approaches. Let us summarize the approaches to evaluation in the research described above.

In terms of its efficiency and suitability to assist experts in the online reputation management task, some approaches are best suited than others. For instance, the models *ON-line LDA* and *Dynamic Topic Model* are able to process

[3] www.blogpulse.com/trends
[4] ilps.science.uva.nl/MoodViews
[5] www.blogscope.net

new documents without re-processing the collection. *Multiscale Topic Tomography*, on the other hand, allows a topic visualization at different granularity levels. However, it is still necessary to define an evaluation framework to compare approaches in terms of accuracy.

Some approaches are simply evaluated over case studies. This is the case of Mei's approach [6] and the *Dynamic topic model* [9]. Ghuhl's model [4], on the other hand, is evaluated against human annotated topic terms; an evaluation method than can not, for instance, be applied to LDA-based models.

The model *Topic over Time* [8] is evaluated with intrinsic clustering metrics according to the KL-divergence between topics; but this methodology is only appropriate to compare similar systems. For instance, in their evaluation the authors obtained evidence about the advantages of including the temporal variable in the model. It is not possible, however, to evaluate heterogeneous systems with intrinsic clustering metrics. For instance, systems based on KL-divergence would be rewarded by this evaluation method. Something similar happens with the evaluation of the *Multiscale Topic Tomography*[11], where the perplexity of the model is compared against the perplexity obtained with other models. In this case, LDA-based models could not be compared with models based on traditional clustering algorithms. Other proposed evaluation metrics focus on extrinsic tasks using topic descriptors, such as multi-document summarization [2].

With these limitations in mind, our goal is to define and apply an automatic evaluation framework enabling comparison between heterogeneous, arbitrary systems, and which is not dependent on cost-intensive manual annotation of data.

3 Evaluation methodology

3.1 System prerequisites

We start from a few prerequisites for topic detection systems in opinion mining:

- **Aggregation:** The system must detect a finite number of topics. Documents will be associated to zero, one or more topics in a discrete or continuous way. The key point is that related documents should share at least one topic.
- **Temporality** In order to analyze the evolution of the reputation of a given entity, the system must reflect differences in topic distribution across time. This implies to show the intensity of topics across time slots.
- **Interpretability** Identified topics should be tagged in a way that is interpretable for the user.
- **Accessibility** For each topic, the corresponding documents must be ranked according to its relevance in the context of the topic.

In this work, we focus on the two first functionalities: "aggregation" and "Temporality". The interesting aspect of these two features is that it is possible to generate automatically a benchmark for evaluation purposes.

3.2 System Output variables

The aggregation functionality requires to infer to what extent each document is related to each topic. This output can be formalized as $P(\theta|d)$, which represents the distribution θ of topics in each document d. For instance, a traditional discrete clustering algorithm would return $P(\theta_i|d) = 1$ if the document d belongs to the cluster associated with the topic θ_i.

Analogously, the temporality function requires an output variable $P(\theta|t)$ representing the distribution of topics in each time slot t. From the perspective of evaluation, a key aspect is that all functionalities must be mutually consistent. In particular, the intensity of topics (temporality) has to correspond with the number of associated documents in the time slot (Aggregation). Therefore, temporality can be inferred from the output $P(\theta|d)$. Assuming that the intensity of topics is proportional to the number of related topic in the time slot, we can state that:

$$P(\theta|t) = \sum_{d \in t} P(\theta|d)$$

3.3 Evaluation measures

Our evaluation methodology is based on two assumptions on the desired behavior of systems:

- Documents with outlinks that point to the same page and documents produced by the same author will tend to be more topically-related than the average.
- It is easier to find highly related documents in the same time slot (say, blog posts in the same week), than separated by long time periods (such as several months).

As most current systems do not rely on this kind of information, it is possible to use it at least for parameter optimization cycles. Some of the systems do employ temporal information, and therefore the second restriction is not totally system-independent. In such cases, however, the improvement obtained by the use of temporal information can still be measured in terms of the first restriction.

The first step to evaluate systems according to these two assumptions consists of defining when the system considers that two documents are related (as for their topics). This is not straightforward, given that systems generate a distribution of weighted topics for each document. We will assume that one topic is enough to consider that two documents are related, but only if both documents focus on this topic. According to this, we define the *Connectedness* of a document pair as:

$$\text{Connectedness}(d1, d2) = \text{Max}_i(\text{Min}(P(\theta_i, d1), P(\theta_i, d2)))$$

Our evaluation metrics will compare the *connectedness* of document pairs in two sets according to the assumptions introduced above. We will call these sets RDP (Related Document Pairs) and NRDP (Non-Related Document Pairs).

RDP consists of document pairs with, for instance, one or more common outlinks, while NRDP consists of documents, conversely, without common outlinks. According to our assumptions, document pairs in RDP should have a higher connectedness, in average, than document pairs in NRDP.

In order to avoid dependencies on scale properties of the distribution $P(\theta|d)$ associated to each system, we will formulate evaluation metrics in a non-parametric way, estimating, for each system s:

$$\text{metric value}(s) = P(\text{Connectedness}(d_r, d_r{}') > \text{Connectedness}(d_n, d_n{}'))$$

where $< d_r, d_r > \in RDP, < d_n, d_n{}' > \in NRDP$.

In other words, the quality of the system is measured as the probability that two documents from the RDP set have a higher topic overlap (according to the system) than two documents from NRDP. Different criteria to form RDP and NRDP lead to different evaluation metrics; we now discuss some examples.

Outlink Aggregation In order to obtain the set of related document pairs (RDP), we assume that two documents are more likely to be related if they share an outlink to the same web page, if this outlink does not appear in other documents (this restriction eliminates frequent outlinks which are not related to the document content, such as links to Facebook).

Author Aggregation As for documents related by a common author, we will simply consider pairs of documents with the same author for RDP and pairs from different authors for NRDP.

Temporality As for temporality, we will assume that is easier to find documents sharing a topic when both documents belong to the same time slot. In particular, we will build RDP with the 100 most related document pairs (according to the system output) which are created in the same week. NRDP is formed by the 100 most related document pairs which are created with a difference of at least three months.

4 Test Case: Iterations in an LDA-based system

To test our evaluation methodology, we have implemented the LDA approach, starting with the algorithm described in [8] and eliminating the temporal variable component – which will be tested in future work –. LDA is a generative process where each document d is associated with a multinomial distribution of topics, and uses Dirichlet distributions as hyperparameters. The model assumes that each document token is associated to a single topic, and therefore the topic distribution in a document would be given by the individual token assignments. The article by Wang and McCallum describes the approach in detail as well as the derivation of the Gibbs sampling.

The algorithm implemented consists of the following steps:

1. Random initialization of each token to some of the k topics.
2. For each token in document d, the topic is updated drawing on the probability $P(z)$ for each topic z. The probabilities are:

$$P(z) = (m_{d,z} + \alpha)\frac{\dot{n_{z,w}} + \beta}{\sum_v^V (n_{z,w} + \beta)}$$

   where $m_{d,z}$ represents the number of tokens in the document d associated to topic z; $n_{z,w}$ represents the number of occurrences of the word w from the corresponding token in topic z, and V is the vocabulary. α and β are two hyperparameters that reflect, respectively, the topic dispersion per word and per document.
3. $m_{d,z}$ and $n_{z,w}$ are updated and then we go back to step 2, for as many iterations as desired.

Implementations known to us use fixed hyperparameters for any word in the vocabulary and for every document. In future work, however, and counting with an automatic evaluation mechanism, we could test whether α should have some relation with the document length.

5 Hypothesis validation

In order to validate our assumptions, we have performed a small experiment which involves manual validation of document pairs.

From our testbed, we have generated 64 random tuples, each consisting of two document pairs: in one pair, both documents share at least one outlink that does not appear in any other document (see Section 3.3); in the other pair, they do not share any outlink. According to our hypothesis, document pairs which share outlinks should be more topically related in average than pairs that do not share outlinks.

For each tuple, we have manually annotated which is the most topically related document pair (sometimes this is not obvious and the tuple is then annotated as undecidable). For 50 tuples (78%), the document pair that share the outlink was more topically related than the other. In 12 cases (19%) it was undecidable, and only in 2 cases (3%) the linked document pair was less topically related than the other.

An analogous process was conducted for the co-authorship criterion, comparing tweet pairs written by the same author with pairs written by different authors. The results over 97 tuples are similar to the previous ones: co-authored tweets are more related in 80% of the cases, while non co-authored tweets are more related only in one occasion (1%).

These results suggest that our assumptions are reasonable for our testbed. This is, of course, just a preliminary result that must be validated with larger manual annotations over different testbeds. Note also that the experimental procedure must be refined, because "undecidable" cases (which are 20% of the assessed samples) might become decidable with a more precise, testbed-specific definition of relatedness.

6 Experiment and Evaluation Results

The goal of our experiment is to test the behavior of our evaluation metrics. As a dataset we use 5,000 tweets and 500 posts from blogs in Spanish containing the term BBVA (an Spanish bank operating in several countries). We have generated a vocabulary excluding stop words. In general, for all the approaches compared, topics detected by the LDA system consist of (i) information about "Liga BBVA", the Spanish Premier Football League, which is sponsored by the bank; (ii) economic information on the bank; (iii) information in languages other than Spanish (such as Catalan); and (iv) topics with unfrequent terms. In general, the granularity of the topics is relatively low. Note that, unlike other related experiments, we are focusing on a single entity, while other approaches cover several totally independent topics.

Table 1 displays the results of *Aggregation* (by outlinks) and *temporality* for LDA over 500 blog posts, fixing certain values of the hyperparameters and for different number of iterations. Note that aggregation goes from 0.41 up to 0.9, reaching a ceiling after 100 iterations. This number might be different when more documents are processed, or for different values of the hyperparameters. In any case, the results show a strong correlation between the number of iterations of the algorithm and the evaluation results; assuming that a higher number of iterations leads to better LDA results, our metric behaves consistently.

Table 1 also shows the *Temporality* obtained for different number of iterations. This metric also increases with the number of iterations, although this behavior is not so stable here. A possible reason is that the assumption that documents which are closer in time should be more related on average is not so valid as for the case of aggregation. Another possible reason is that this measure is estimated on the 100 most related document pairs only, while aggregation is computed on a larger set of samples.

Iterations	Agregation (by outlinks)	Temporality
1	0.41	0.42
5	0.60	0.35
10	0.76	0.6
20	0.86	0.57
50	0.89	0.60
100	0.90	0.61
200	0.90	0.61
500	0.90	0.65
1000	0.91	0.71
2000	0.90	0.72

Table 1. Evaluation results for 500 blog posts, 10 topics, $\alpha = 1$, $\beta = 0.1$ and different number of iterations

Table 2 shows the results on 5,000 tweets, this time measuring aggregation by author. Again, the metric values seem to stabilize around 100 iterations, and they show a clear correlation with the number of iterations.

Iterations	Agregation (by author)
1	0.47
5	0.55
10	0.61
20	0.72
50	0.76
100	0.78
200	0.79
500	0.8

Table 2. Evaluation results for 5000 tweets, 10 topics, $\alpha = 1$ and $\beta = 0.1$

Another variable that can be analyzed with our evaluation methodology is the effect of different values for the hyperparameter α. Table 3 shows that α does not have a strong effect on the results. In fact, the maximum seems to be around $\alpha = 1$. This implies that, in general, documents tend to be centered around one single topic. This is perhaps due to the low granularity of the topics produced in our experiment.

alpha value	Agregation (by outlinks)	temporality
0.1	0.89	0.64
1	0.9	0.66
5	0.9	0.67
10	0.89	0.66
20	0.88	0.74
50	0.84	0.62

Table 3. Evaluation results for 500 blog posts, 2000 iterations, 10 topics, $\beta = 0.1$ and different α values

Finally, we have studied the effect of the number of topics on the results of the evaluation. Is it possible that LDA, in this context, reaches a more adequate topic granularity by increasing their number? Table 4 shows the results obtained for 500 blog entries, 2000 iterations and $\alpha = 1$. Note that, although there is some positive effect when increasing the number of topics, it is not as clear as in previous experiments.

Number of topics	Agregation (by outlinks)	temporality
5	0.85	0.67
10	0.9	0.73
15	0.92	0.74
20	0.92	0.68
40	0.93	0.71
50	0.93	0.7

Table 4. Evaluation results for 500 blog posts, 2000 iterations, $\alpha = 1$, $\beta = 0.1$ and a variable number of topics

7 Conclusions

In this work we have proposed an early version of an automatic evaluation methodology which permits the optimization of topic extraction models for online reputation management using external information not employed by the system. In a preliminar experiment using blog entries and tweets for a bank, we have been able to observe quantitative effects such as a little influence of the α hyperparameter on the final results, the number of iterations which lead to stable results for LDA, or the effect produced by the number of topics.

Our evaluation methodology has still unresolved issues: we do not know yet to which extent the selection of the RDP and NRDP sets bias the results (they are, after all, just a small sample of the full test set, with very precise characteristics). We also need to revise the "temporality" measure to obtain more stable results in our experimental framework.

In any case, the methodology provides a way of testing hypothesis not yet evaluated quantitatively in other studies, such as the effect of including a temporal variable in the model, the possibility of processing time slots independently, the effects of structuring topics hierarchically, etc.

References

1. Hurst, M.: Temporal text mining. In: Proceedings of the AAAI Spring Symposia on Computational Approaches to Analyzing Weblogs. (2006)
2. Subasic, I., Bettina, B.: From bursty patterns to bursty facts: The effectiveness of temporal text mining for news
3. Brooks, C.H., Montanez, N.: Improved annotation of the blogosphere via autotagging and hierarchical clustering. In: WWW '06: Proceedings of the 15th international conference on World Wide Web, New York, NY, USA, ACM Press (2006) 625–632
4. Gruhl, D., Guha, R., Liben-Nowell, D., Tomkins, A.: Information diffusion through blogspace. In: WWW '04: Proceedings of the 13th international conference on World Wide Web, New York, NY, USA, ACM (2004) 491–501
5. Chi, Y., Tseng, B.L., Tatemura, J.: Eigen-trend: trend analysis in the blogosphere based on singular value decompositions. In: CIKM '06: Proceedings of the 15th

ACM international conference on Information and knowledge management, New York, NY, USA, ACM Press (2006) 68–77
6. Mei, Q., Liu, C., Su, H., Zhai, C.: A probabilistic approach to spatiotemporal theme pattern mining on weblogs. In: WWW '06: Proceedings of the 15th international conference on World Wide Web, New York, NY, USA, ACM Press (2006) 533–542
7. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. Journal of Machine Learning Research **3** (2002) 2003
8. Wang, X., McCallum, A.: Topics over time: a non-markov continuous-time model of topical trends. In: KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, New York, NY, USA, ACM (2006) 424–433
9. Blei, D., Lafferty, J.: Dynamic topic models. In: Proceedings of the 23rd international conference on Machine learning, ACM New York, NY, USA (2006) 113–120
10. AlSumait, L., Barbara, D., Domeniconi, C.: On-line LDA: adaptive topic models for mining text streams with applications to topic detection and tracking. In: Proceedings of the 2008 Eighth IEEE International Conference on Data Mining, IEEE Computer Society Washington, DC, USA (2008) 3–12
11. Nallapati, R.M., Ditmore, S., Lafferty, J.D., Ung, K.: Multiscale topic tomography. In: KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, New York, NY, USA, ACM (2007) 520–529

Index