Workshop on Analysis of Complex NEtworks

at ECML PKDD 2010

ACNE 2010

September 20, 2010

Barcelona, Spain
Many real world data collections and information structures are dynamic and rich in information. Represented as complex networks, a mesh of interconnected information entities, allows to get the most out of them. Analysing complex networks regarding non-trivial aspects such as information propagation, evolution, or community discovery and discovering or even creating new connections requires powerful tools for supporting visualisation, interaction and mining of and with the underlying data. While many graph mining tools have been developed there is a lack of techniques able to tackle real-world networks which are typically multi-dimensional, heterogeneous, and/or dynamic. Furthermore, sophisticated tools supporting the interaction with huge networks are virtually non-existent. Besides refined mining and aggregation methods there is a need for modelling and sophisticated visualisation methods which take the limited amount of information into account a user can process.

The aim of the workshop is to bring together pioneering researchers focussing on the analysis of complex networks, and thus intensify the exchange of ideas between different research communities to foster devising tools for creation, analysis, and visualisation of complex networks. The workshop focuses especially on researchers working on methods for mining, learning and analysis of (dynamic) networks, the representation and semantics of complex networks, and visualisation methods as well as user interface design.

Out of the submitted papers, we have selected two extended abstracts and eight full papers. Each of these papers deals with some aspects of the analysis of complex networks. All the papers and presentations will be available on the workshop website http://kdd.isti.cnr.it/acne/.

We are also proud to welcome our invited speaker: Debora Donato, from Yahoo! Research. We gratefully thank her for having accepted our invitation to give a talk at the workshop.

The organizers would like to acknowledge the support from the BISON project (FP7-ICT-2007-C FET-Open, contract no. BISON-211898). We also would like to thank the reviewers and the program committee members for their help. Finally, we thank the ECMLPKDD 2010 organising committee for their support in the organization of this workshop.

Michele Berlingerio, Björn Bringmann, and Andreas Nürnberger, August 2010.
Workshop Organisation

Editors

*Michele Berlingerio*
KDD Laboratory, ISTI-CNR Pisa, Italy

*Björn Bringmann*
Katholieke Universiteit Leuven, Belgium

*Andreas Nürnberg*
Otto-von-Guericke-University Magdeburg, Germany

Program Committee

Michael Berthold, University of Konstanz, Germany
Christian Borgelt, European Center for Soft Computing, Mieres, Spain
Toon Calders, Eindhoven University of Technology, The Netherlands
Michele Coscia, KDD-Lab - University of Pisa, Italy
Fabrizio Costa, Katholieke Universiteit Leuven, Belgium
Luc De Raedt, Katholieke Universiteit Leuven, Belgium
Debora Donato, Yahoo! Research, Silicon Valley, USA
Werner Dubitzky, University of Ulster, United Kingdom
Fosca Giannotti, KDD-Lab - ISTI-CNR, Pisa, Italy
Andreas Hotho, University of Kassel, Germany
Laks Lakshmanan, University of British Columbia, Vancouver, Canada
Nada Lavrač, Jožef Stefan Institute, Slovenia
Sune Lehmann, Harvard University, USA
Jure Leskovec, Stanford University, USA
Trevor Martin, University of Bristol, United Kingdom
Anna Monreale, KDD-Lab - University of Pisa, Italy
Siegfried Nijssen, Katholieke Universiteit Leuven, Belgium
Dino Pedreschi, KDD-Lab - University of Pisa, Italy
Josep Maria Pujol Serra, Telefonica Research, Barcelona, Spain
Fabrizio Silvestri, HPC-Lab - ISTI-CNR, Pisa, Italy
Chaoming Song, Northeastern University, Boston, USA
Shirish Tatikonda, Ohio State University, Columbus, USA
Hannu Toivonen, University of Helsinki, Finland
Yana Volkovich, Barcelona Media, Barcelona, Spain
Philip Yu, University of Illinois at Chicago, USA
Table of Contents

High Throughput Network Analysis ........................................... 5
Sumeet Agarwal, Gabriel Villar, and Nick Jones

Node Similarities from Spreading Activation .......................... 8
Kilian Thiel and Michael R. Berthold

Link and Node Prediction in Metabolic Networks with Probabilistic Logic 10
Angelika Kimmig and Fabrizio Costa

Exploiting Network Structure in Enhancing Diffusion of
Complex Contagions .............................................................. 20
Chris J. Kuhlman, Madhav V. Marathe, S. S. Ravi, and Daniel J. Rosenkrantz

Cross-Domain Literature Mining through Outlier Document and Bridging Concept Detection .................................................. 35
Nada Lavrač, Borut Sluban, and Matjaž Juršič

Constraint-Based Mining of Sets of Cliques Sharing Vertex Properties .... 48
Pierre-Nicolas Mougel, Marc Plantévit, Christophe Rigotti, Olivier Gandrillon, and Jean-François Boulicaut

Heuristic best-first search in separation of interleaved Web sessions .... 63
Marko Poženel, Viljan Mahnič, and Matjaž Kukar

Frequent Bisociative Term Set Mining in Textual Data .................. 75
Marc Segond and Christian Borgelt

Unfolding network communities by combining defensive and offensive label propagation ......................................................... 87
Lovro Šubelj and Marko Bajec

Applications of stochastic relational processes ......................... 105
Ingo Thon, Niels Landwehr, and Luc De Raedt
High Throughput Network Analysis

Gabriel Villar\textsuperscript{1,2,3}, Sumeet Agarwal\textsuperscript{1,2}, and Nick S Jones\textsuperscript{2,4,5}

\textsuperscript{1} Systems Biology Doctoral Training Centre, University of Oxford, Oxford OX1 3QD, United Kingdom
\textsuperscript{2} Department of Physics, University of Oxford, Oxford OX1 3PU, United Kingdom
\textsuperscript{3} Department of Chemistry, University of Oxford, Oxford OX1 3TA, United Kingdom
\textsuperscript{4} Oxford Centre for Integrative Systems Biology, University of Oxford, OX1 3QU, United Kingdom
\textsuperscript{5} CABDyN Complexity Centre, University of Oxford, Oxford OX1 1HP, United Kingdom

1 Introduction

Gene regulatory systems, metabolic pathways, neuronal connections, food webs, social structures and the Internet are all naturally and widely represented as networks. However, it is not always clear how such a network representation aids the understanding of its real-world counterpart. It may be that abstracting a real-world system as a network discards all of the relevant information, but this seems unlikely for such a high-dimensional representation. Here, we presume that there is some valuable information encoded in the network; the problem is simply to find it. To learn about networks of any significant size it is generally necessary to characterise them by summary descriptions, which we will refer to as metrics.

A great variety of metrics exist in the literature, but studies that aim to characterise a particular network typically employ a small subset of these, and the choice of metrics is not systematic. It is typical for a new metric to be introduced with comparison to only a few existing ones. The lack of a systematic comparison makes it difficult to know which metrics give genuinely novel information about a network, and which might be redundant. Efforts to address this have recently been made \cite{2}, but it remains true that there is as yet no systematic program for characterising network structure \cite{7} that can be used to compare the diverse ways in which networks are analysed. We attempt to introduce such a framework, in the form of a matrix whose rows correspond to networks, and columns to metrics; we term this the data matrix. In this paper we demonstrate some applications of this approach to feature selection and classification tasks on networks.

2 Materials and Methods

2.1 Networks

We collected about 1,200 real networks from a wide variety of publicly available sources. These included several types of biological networks, social networks, computer networks and miscellaneous others. The networks ranged in size from a few tens of nodes upto tens of thousands of nodes.

2.2 Metrics

We included about 60 base metrics taken from the literature. Whilst some metrics, such as the diameter, are simple scalars, others return a series of numbers: for example, the degree distribution returns a number for each degree in the network. In order add such vector metrics to our data matrix, we generated a number of summary statistics of these distributions. Additionally, we include graph clustering or community detection \cite{3,8} metrics: these return a partition of the network into subnetworks, which can also be summarised in various ways. In sum, the combination of simple scalar metrics and summary statistics gives us about 400 features in our data matrix.

3 Selected Results

To demonstrate the utility of this approach for feature selection, we are looking at the problem of detecting phylogenetic signal in biological networks. Rather than just grouping data points into independent classes, we would like
to take into account the entire structure of evolutionary relationships between species, which may be represented by means of a phylogenetic tree. Given such a tree, the objective is to find features of network structure that co-vary with the phylogeny. We are currently working on this using ideas from the area of phylogenetic comparative methods [1, 5, 6]. As a rough preliminary step, we obtained a set of 43 metabolic networks [4] and grouped them at the leaves of a highly simplified phylogeny. Each network was represented by its feature vector, and features were ranked based on information gain at each of the three branching points in our phylogeny. We found that features based on closeness, a measure of node centrality, were amongst the most informative ones at all of the branching points. This suggests that closeness may be a biologically relevant network property, and it should be of interest to study this in greater detail.

As an example to show how unsupervised learning can be used for network characterisation, we took a set of 192 networks from a wide range of disciplines and carried out principal component analysis (PCA), utilising a set of 433 metrics. The results are shown in Figure 1. We see that even in this 2-dimensional mapping, certain kinds of networks seem to fall into very cohesive groupings, suggesting that our feature vectors are capturing functionally important properties of those systems.

Fig. 1. Results of PCA on a set of 192 networks, using 433 features. The two largest principal components are shown.

4 Discussion

In some ways, the approach taken here is complementary to standard perspectives in network science. When a new metric is introduced in the networks literature, it may be motivated by an expectation of what aspects of a network it will capture, or by some distinguishing feature of its calculation. Similarly, new network models are assessed by how closely they match particular metrics. Here, we simply apply all of the available metrics to a set of networks, and use the resulting data structure to explore the networks or metrics in an unprejudiced manner. This framework as a way of systematically comparing metrics should be valuable for both exploratory network analysis, and for finding the best way to answer a particular question in a data-driven manner. Our early results based on relatively simple models and metrics for capturing network structure suggest that our framework may serve as a powerful tool for feature selection and network characterisation tasks in a number of different settings. It continues to be work in progress, but we hope that once complete, public distribution of the software and database built for this project will benefit the community and see new applications of the framework.
References

Node Similarities from Spreading Activation

(Extended Abstract)

Kilian Thiel and Michael R. Berthold

Nycomed Chair for Bioinformatics and Information Mining
University of Konstanz, Germany

{Kilian.Thiel, Michael.Berthold}@uni-konstanz.de

There is a growing amount of datasets that can be represented as networks, for instance biomedical networks, text networks, or social networks. Many of these heterogeneous datasets can be assembled into bigger networks e.g. biocative information networks [1,2], in order to facilitate insights across different domains. With the increase of data, on the one hand the interest in the analysis of such networks increases, on the other hand the network structures become more and more unclear, complex, and unpredictable.

To determine the global status of a node in a network different kinds of centrality indices can be computed [3] such as degree centralities, hubs and authorities and many more. By comparing the centrality values, nodes can be found which have a similar status in the global network.

Besides the global status of nodes in a network, it can be interesting as well to search for nodes with a similar status in their local neighborhood, for instance proteins, in biomedical networks, that play a similar role in their metabolic pathways. Such nodes are similarly connected to their local (direct and indirect) neighborhood. The neighborhoods do not necessarily have to overlap, meaning that the nodes can be located far apart in the graph. To compare the status of nodes in their neighborhood, we define a structural node similarity, called signature similarity. Node pairs with a higher signature similarity value are more similarly connected to their corresponding neighborhood.

To compare and understand the status of two nodes, as well as the structures of their neighborhoods, it is furthermore necessary to extract the subgraphs of their dense direct and indirect neighborhoods, namely communities. Nodes which are located very close in the graph and belong to the same community share in large part their direct and indirect neighborhood [4–6]. We define a second node similarity, called activation similarity, measuring the closeness of two nodes by their direct and indirect neighborhood overlap. Based on this similarity the closest nodes according to a given node can be determined and its dense surrounding subgraph can be extracted.

Both kinds of node similarities are derived from spreading activation [7] processes, one started from each node in the network. The activation similarity is based on the comparison of activation vectors. The signature similarity is based on the comparison of relative change of activation over time (velocity), yielding a structural similarity.
We applied both methods to the Schools-Wikipedia\(^1\) (2008/09) dataset and preliminary results are very encouraging. For example the node of Larry Page is structurally similar to that of Linus Torvalds. By comparing their communities this analogy could be confirmed manually. Both nodes are not the most central ones in their community but are directly connected to them (Google and Internet, Linux and Unix, respectively). Furthermore they are similarly connected to the rest of their community. The experiments suggested that the combination of these two kinds of similarities could be a promising tool in the area of extraction of *bisociations* based on similar graph substructures \([2]\) as well.

**Acknowledgment**

We would like to thank the members of the EU Bison project for many fruitful discussions during the development of BisoNets. The work presented in this paper was supported by a European Commission grant under the 7th Framework Programme FP7-ICT-2007-C FET-Open, project no. BISON-211898.

**References**


\(^1\) [http://schools-wikipedia.org/](http://schools-wikipedia.org/)
Link and Node Prediction in Metabolic Networks with Probabilistic Logic

Angelika Kimmig and Fabrizio Costa

Departement Computerwetenschappen, K.U. Leuven
Celestijnenlaan 200A - bus 2402, B-3001 Heverlee, Belgium
{angelika.kimmig, fabrizio.costa}@cs.kuleuven.be

Abstract. Information on metabolic processes for hundreds of organisms is available in public databases. However, this information is often incomplete or affected by uncertainty. Systems capable to perform automatic curation of these databases and capable to suggest pathway-holes fillings are therefore needed. Using ProbLog, a simple yet powerful extension of the logic programming language Prolog with independent random variables, we start to investigate two fundamental problems concerning automatic metabolic networks curation, namely link prediction and node prediction.

1 Introduction

We are nowadays capable of representing organism-wide metabolic processes. In fact there exist collections of metabolic networks for several hundreds of organisms (e.g. the Kyoto Encyclopedia of Genes and Genomes (KEGG) [1] or the BioCyc database [2]) where relations between genes, enzymes, reactions and chemical compounds are available. The knowledge that we have of these relations is however incomplete (most annotation efforts fail to assign functions to 40-60% of the sequences [3]) and is affected by uncertainty (wrong EC number assignment, incomplete annotation (e.g. only one function of a multidomain protein) or nonspecific assignment (e.g. to a protein family)). Systems capable to perform automatic curation of these databases and capable to suggest pathway-holes fillings are therefore needed. For this purpose one can make use of information on related organisms and use evidence based not exclusively on homology searches, but also on genomic and/or functional context. This raises the problem of how to integrate heterogeneous and uncertain sources of information in a principled way. Although systems for reconstructing pathways from relevant gene sets [4] and filling pathway-holes [5] are known in literature, they do not offer sufficient flexibility when new additional sources of information become available or, more importantly, in case one needs to change the set of queries involved in the solution of a specific task.

We have studied an approach that satisfies these flexibility requirements by representing metabolic networks in a probabilistic logical framework. In this way background knowledge affected by uncertainty can be easily included, and we can
obtain an answer to several key questions performing probabilistic inference in a principled manner. More specifically, we use ProbLog [6], a simple yet powerful extension of the logic programming language Prolog with independent random variables in the form of probabilistic facts.

In this work we start to investigate some fundamental problems concerning automatic metabolic networks curation, namely: 1) \textit{link prediction}, i.e. the correction of the link strength between a gene and an enzyme, and 2) \textit{node prediction}, that is, whether the existence of a certain enzyme (and hence of an unknown gene) has to be hypothesized in order to maintain the contiguity of a pathway.

2 The Probabilistic Logic Environment: ProbLog

In contrast to propositional graphical models (such as Bayesian Networks), connections between random variables in ProbLog can be specified on the first order level, thus avoiding the need of explicitly grounding all information a priori, achieving therefore a higher abstraction and flexibility in the queries specification.

More formally, a \textit{ProbLog program} $T$ consists of a set of labeled facts $p_i :: c_i$ together with a set of definite clauses encoding \textit{background knowledge} (BK). Each ground instance of such a fact $c_i$ is true with probability $p_i$, where all probabilities are assumed mutually independent. The program thus naturally defines a probability distribution

$$P(L|T) = \prod_{c_i \in L} p_i \prod_{c_i \notin L \cap T} (1 - p_i)$$

over logic programs $L \subseteq L_T = \{c_1, \cdots, c_n\}$. The \textit{success probability} of query $q$ is then defined as

$$P_s(q|T) = \sum_{L \subseteq L_T} P(q|L) \cdot P(L|T)$$  \hspace{1cm} (1)$$

where $P(q|L) = 1$ if there exists a $\theta$ such that $L \cup BK \models q\theta$, $P(q|L) = 0$ otherwise. It thus corresponds to the probability that $q$ is provable in a randomly sampled logic program. To calculate success probabilities, ProbLog 1) constructs a DNF formula representing all proofs of the query, and 2) uses Binary Decision Diagrams (BDDs) [7] to efficiently calculate the probability of this formula being true in a randomly sampled program. The probability of a DNF formula cannot be obtained directly from the probabilities of the different proofs, as each possible world can allow for multiple proofs. This problem is also known as the \textit{disjoint-sum-problem} or the two-terminal network reliability problem, which is \#P-complete [8]. BDDs offer a way to tackle the problem without the need to enumerate all possible worlds by compactly representing a Boolean formula as an acyclic directed graph. A DNF formula could naively be encoded as a full Boolean decision tree where each layer corresponds to one probabilistic fact and each leaf is labeled with the truth value of the query in the world given by the
truth value assignments on the path to this leaf. Using BDDs is similar in spirit, with two important differences. First, redundancies in the tree are exploited to obtain compact representations, and second, BDDs are built by combining BDDs for subformulas, thus avoiding the need to construct the entire tree. Probabilities can then be calculated by a single bottom-up pass through the final structure; we refer to [6] for more details.

3 Method

3.1 Metabolic Networks

We represent the knowledge about metabolic networks in a probabilistic logical framework. To this end, we identify the main entities involved in the problem and encode all relations between them quantifying the uncertainty of each relation with an associated probability value (see Figure 1). The entities that we consider are: organisms, genes, enzymes, reactions, compounds (also called metabolites) and pathways; the relationships considered are: organisms are phylogenetically related to other organisms; enzymes are related to enzymes in the enzyme functional hierarchy given by the Enzyme Commission number (EC number) system [9]; genes are related to genes via the ortholog relationship (see further in the text); genes are related to the organisms they are part of; reactions are related to the compounds they require as substrate and to those that are produced; genes are related to the enzymatic function of the protein that they code for; enzymes are related to the reactions they catalyze; and finally pathways are collections of related reactions. Currently only the gene-enzyme relation is treated probabilistically while all the other relations are assumed to be known with certainty and are derived from the KEGG Database [1]. Note that in principle all relations are of the type many-to-many although in practice a gene is almost always associated to a single enzyme, which in turn catalyzes almost always a single reaction (see Figure 1). Informally, a metabolic network contains information on the set of genes that belong to specific organisms and how these code for proteins, called enzymes, that are responsible for specific reactions involving the transformation of one compound into another. An organism is thus capable to perform certain related sets of reactions (semantically grouped under a single pathway concept) in order to produce and transform sets of metabolites, only if the organism can express the enzymes needed to catalyze those reactions.

3.2 Learning Task

Given the metabolic information about a set of organisms we identify two main problems of interest relevant for the concept of automatic network curation: 1) link prediction, where we estimate the probability associated to a given set of relations on the basis of an initial guess, so to increase the consistency with respect to the information on related organisms; and 2) node prediction, where we introduce specific nodes in order to best fill gaps in the pathway of interest.
More in detail, we work in the following setting: we are given information about a new organism consisting of a set of genes and their associated functions (i.e. the enzyme they code for); this information is understood affected by uncertainty and is a preliminary approximation that needs to be refined to increase consistency; as background knowledge we are given information on the metabolic network for a large set of organisms; furthermore we are given two similarity notions: the first one is between the test organism and other organisms (obtained from the phylogenetic tree) and the second is between the genes in the test organism and genes in other organisms. This latter information is available in KEGG and is obtained via an heuristic method that determines an ortholog cluster identifier in a bottom-up approach. In this method, each gene subgroup is considered as a representative gene and the correspondence is computed using bi-directional best hit (BBH) relations obtained from the KEGG SSDB database which stores all-vs-all Smith-Waterman similarity scores [10]. For efficiency reasons, both similarity scores are currently thresholded and binarized: in practice, two genes are linked via the ortholog relation only when each one is ranked in the top $k$ most similar genes of the other and when the similarity between the two exceeds a pre-specified threshold.

**Link prediction task:** In order to re-estimate the strength of a gene-enzyme relation, we consider evidence coming from different types of substructures in the network (see Figure 2 left and Figure 3) and use the ProbLog inference engine to compute the associated probability value. In principle we prefer evidence coming

---

**Fig. 1.** Part of KEGG metabolic network used. The number in the node shape is the cardinality of the element set. The number on the edge is the average ± standard deviation number of relations between the element at the starting endpoint and the elements at the final endpoint of the edge. Dashed elements represent information present in KEGG but not currently used.
Fig. 2. Graphical representation of the portion of metabolic network used to obtain evidence for the link prediction task (left) and the node prediction task (right).

from more complex substructures but in practice this information is not always available. The reason for this phenomenon is the partial knowledge that we have of the metabolic network: a) not all genes of a test organism have an initial associated function; b) not all genes have known orthologs; c) not all reactions are known in a given pathway. If the database does not contain information to completely match a complex query against, this query will simply fail as it cannot be proven. Hence, it does not provide any information, or, in other words, contributes a probability of 0. In these cases we resort to increasingly simpler queries in a fashion similar in spirit to the interpolation techniques employed in computational linguistic\textsuperscript{1}. Finally we integrate information coming from the increasingly complex queries via linear model whose coefficients are learned under a supervised scheme.

In order of complexity we consider: 1) evidence of the strength of gene-enzyme relation either known a-priori or computed by a predictive system (Figure 3 left), this corresponds to the initial estimate embodied as a link between the gene and the enzyme; 2) evidence coming from paths, that is, the probability of a path that involves the gene-enzyme link under consideration (Figure 3 middle); and 3) evidence coming from a complex subgraph, that is the probability of a network portion that involves both the gene-enzyme link and links of ortholog genes in related organisms (Figure 3 right). ProbLog allows us to specify the characteristics of these substructures at an intensional level. In particular we require 2) to be a path that traverses in order the following selected types of entities: gene, enzyme, reaction, compound, (reaction-compound)*, reaction, enzyme, gene. The intended meaning of the star notation here is that the path is only allowed to follow further reaction-compound links if the current reaction does not have an

\textsuperscript{1} When employing \textit{n-gram} models, a common practice is to assess the probability of complex n-grams using the frequency counts of smaller n-grams that are more likely to occur in (small) datasets.
enzyme associated in the database. This latter condition is motivated by both computational efficiency issues (i.e. we do not consider all possible paths but only the shortest ones) and the desire to favor paths that make use of information relevant to the test organism. In words: we consider linear chains that originate in one gene of the test organism and end up in another gene of the same organism traversing the enzyme-reaction network relevant to a specific pathway. The subgraph for case 3) is obtained considering paths of type 2 with the addition of two extra paths that originate from genes in the test organism, traverses ortholog genes and end up in the enzymes of interest at both ends of the original path of type 2. The ratio here is to prefer evidence that is consistent with the information on similar genes in different organisms.

**Node prediction task:** Here we compute the probability of an enzyme and adapt the structures we use to provide evidence in the following way (see Figure 2 right): first of all, note that we cannot consider structures of type 1), that is an a-priori estimate as we work precisely under the assumption that no information is known on the existence of a gene associated to a specific enzymatic activity; instead we consider the average association strength of given enzyme with any known gene present in related organisms; for the more complex queries we consider paths similar to those of type 2) where the initial gene-enzyme link is removed and is substituted by a gene in some other related organism, but where we still require the path to end in a gene that is known to belong to the test organism; finally for the most complex query we consider subgraphs similar to 3) where the initial gene-ortholog gene-enzyme chain is replaced by a gene-enzyme relation between a gene belonging to a related organism; for the other endpoint of the path we use the information available about genes that are orthologs of test genes.

**Learning task:** In both the link and node prediction setting we introduce queries that can be answered in multiple ways (e.g. there is more than one path that starts from a given initial gene and ends in another gene of the same organism). Each solution comes equipped with an associated probability of being true and each contributes evidence to the overall probability to satisfy the query.
predicate. Since the various solutions are not independent we cannot derive the final probability simply by summing up all the returned probabilities. This is an instance of the disjoint-sum-problem, which in ProbLog is tackled by resorting to BDDs as explained in Section 2.

Finally we use the results returned for each different query type to answer the main questions: what is the probability of a specific gene of a test organism to be associated to a specific enzyme? or what is the probability of a specific enzyme to belong to a pathway for a given test organism? We compute these probabilities via a linear model whose parameters are learned using ProbLog’s gradient-descent approach to parameter learning [11]. Given a set of queries with associated target probabilities, this method uses standard gradient descent to minimize the mean squared error (MSE) on the training data. To do so, it exploits the BDDs used in ProbLog inference to also calculate the gradient.

The idea behind the linear model is to learn how to adapt to the level of missing information in the network: when predicting the association strength with an enzyme that is embedded in a network region where few reactions are known it is better to trust the prior estimate with respect to more complex queries since they will mainly fail over the poorly connected reaction network; analogously when ortholog genes are known for a given enzyme, the evidence from the more complex queries becomes compelling. In summary, we adapt to the unknown local quality of the network by learning the relative importance of each query for the final answer, and we do this by training a model on related organisms.

4 Experimental Setup

4.1 Noise Model

Instead of working with a specific gene function predictor we study the curation/reconstruction capacity of the proposed system perturbing the knowledge of the true function of a gene in a specific and controlled way. Since the enzymatic functions can be arranged in a hierarchical ontology [9] we posit that we can relate the topological distance in the ontology tree to the functional distance, i.e. the closer two enzyme nodes are in the hierarchy the more similar their functions. Under this assumption we build a noise model described by the following parameters: 1) $s$ fraction of affected genes; 2) $k$ number of noisy gene-enzyme links added per gene; 3) $\sigma_{EC}$ parameter controlling the size of the neighborhood where to randomly sample the additional noisy gene-enzyme links; 4) $\sigma_N$ parameter controlling the quantity of noise added to the gene-enzyme relationship probability estimate. We then proceed as follows (see Figure 4): given an organism we select a fraction $s$ of its genes; for each gene we add $k$ extra links to randomly sampled nearby enzymes; here the informal notion of a metric is formally defined as a normal probability distribution (of selecting an enzyme) $N(0, \sigma_{EC})$ and support over the topological distance induced by the ontology (i.e. the length of the shortest path between the leafs containing the two enzymes in the tree structured ontology); finally the strength of the link between the gene
Fig. 4. Noise model: the E.C. hierarchy induced metric notion (i.e. topological distance between nodes) is used as support for the perturbed enzymatic function. The hypothetical true enzyme is marked with a double line. In the example a gene is associated to an incorrect enzymatic activity with probability 0.52 and to the correct one with probability 0.4.

and the randomly selected enzyme is computed as the probability of selecting the enzyme with additional $N(0, \sigma_N)$ noise: in this way enzymes that are less related to (i.e. more distant from) the true enzymatic function of the original gene receive on average a smaller probability.

4.2 Experimental Results

In the experiments reported here, we focus on the Pyruvate metabolism pathway for the Escherichia coli UTI89 test organism. We perturb the true relationships with $k=5$ extra links for $s = 50\%$ of genes. The probability estimate of the gene-enzyme relationship receives additional noise from $N(0, \frac{1}{8})$.

We use default settings in our experiments and run learning for at most 50 iterations, stopping earlier if the MSE on the training data does not change between two successive iterations. Training data is generated from the other organisms with the same parent in the organism hierarchy as the test organism, and target probabilities are set to $1.0$ for positive and $0.0$ for negative examples, respectively.

In the link prediction setting, positive examples are real gene-enzyme links, while negative ones are the ones added by the noise model where no real one is known between these entities. We use the three queries depicted in Figure 3. We measure the area under the precision-recall curve.

When using the initial (perturbed) estimate for the gene-enzyme link we achieve an AUCPR of 0.69. If we use only the most complex query (type 3) we
increase to 0.74, but when we learn the logistic model over all queries we achieve 0.80. Note that simply learning a fixed mixture of experts for the whole organism (i.e. not modeling the dependency on the enzyme) we do not improve over the initial 0.69 result as for this particular test organism, it is better to resort on average to the most simple query.

In the node prediction experiment, we adopt an enzyme level leave-one-out design. From the background knowledge we retrieve all the enzymes that do not have an associated gene in the test organism. We remove all enzymes in turn and we measure the precision at one, that is the fraction of times that the missing enzyme is ranked in first position as the most probable among all the missing enzymes.

The set of training examples is the set of all pairs of training organisms (as before) and enzymes appearing in the pathway for organisms different from the test organism. Such a pair is considered positive if the enzyme appears in the organism’s pathway, and negative else.

We use the query described in Section 3 both with and without ortholog information, as well as a basic query that predicts each enzyme with the average probability of a gene-enzyme link involving this enzyme in one of the training organisms. In this experiment we achieve a precision at one of 0.66 over 35 possible enzymes (i.e. the baseline random guessing precision at one would be 0.03).

5 Conclusions

We have started tackling the problem of automatic network curation by employing the ProbLog probabilistic logic framework. This choice has allowed us to:

a) represent the knowledge about the metabolic network even when affected by uncertainty, and
b) express complex queries to extract evidence for the presence of missing links or nodes in an abstract and flexible way. Initial experimental evidence shows that we can effectively recover missing or inconsistent information. Future work includes the integration of concrete gene function predictor and the development of novel queries that make use of additional sources of information such as the gene position in the genome or the co-expression of genes in the same pathway from medical literature abstract analysis.

Acknowledgments

A. Kimmig is supported by the Research Foundation-Flanders (FWO-Vlaanderen). This work is partially supported by the GOA project 2008/08 Probabilistic Logic Learning and by the European Commission under the 7th Framework Programme, contract no. BISON-211898.
References

Exploiting Network Structure in Enhancing Diffusion of Complex Contagions

Chris J. Kuhlman¹, Madhav V. Marathe¹, S. S. Ravi², and Daniel J. Rosenkrantz²

¹ Virginia Bioinformatics Institute, Virginia Tech, Blacksburg, VA 24061, USA. Email: {ckuhlman,akumar,mmarathe}@vbi.vt.edu
² Computer Science Department, University at Albany – SUNY, Albany, NY 12222, USA. Email: {ravi,djr}@cs.albany.edu

Abstract. Enhancing the diffusion of complex contagions in social networks is important in applications such as product marketing. We study the diffusion process for complex contagions with a focus on understanding the role played by the network structure in fostering diffusion in large networks. We present theoretical bounds that relate a dynamic property of a network (namely, the maximum number of nodes to which the contagion spreads) to a static and efficiently computable parameter of the network (namely, the size of a specific subnetwork). These bounds allow us to develop a heuristic for choosing a subset of nodes where the diffusion process can start so as to spread to a large part of the network. We use this heuristic to carry out an empirical study of the diffusion process for complex contagions on synthetic networks as well as two practical social networks (namely, epinions and slashdot).

1 Introduction

There have been many efforts to mine data from different sources, particularly Internet sources, to construct networks or acquire data that describe social behaviors [16,20,22,23,29]. Each of the processes of collecting information, making decisions about how to parse it, and generating networks is difficult [16], and choices must often be made [11, 22, 23, 31] in the absence of well-defined guidelines. Analyzing social networks to understand social behavior is important in several applications including product marketing and controlling an epidemic, and the quality of these analyses is dependent on the quality of the networks. A commonly used model to study this behavior is that of diffusion processes in social networks (see for example, [3,4,15]).

In this work, a social network consists of nodes that represent agents (e.g., people) and edges that represent interactions between neighboring agents. We are interested in diffusion processes where some entity (sometimes called a contagion), such as information, a fad, or influence, is transferred among agents via the edges. The local interactions between agents and their neighbors dictate the global behavior of the system. A diffusion process is started by seeding or activating a subset of the nodes; that is, these seed nodes initially possess the
contagion, which may subsequently spread through the network. As diffusion progresses, nodes change their states. In the case of information propagation, a node that does not possess the information is in a certain state (say, 0); when it acquires the information, its state changes (say, to 1).

Complex contagion and simple contagion describe diffusion processes that require input from, respectively, multiple neighbors and a single neighbor in order for an agent to change its state or behavior. Simple and complex contagions are also identified as $t = 1$ and $t > 1$ threshold models, respectively, where $t$ is the threshold value. (A precise definition of the model is given in Section 2.1.) Disease propagation is often considered a simple contagion, while many social phenomena such as fads, adoption of technological innovations, and mob behavior are considered to be complex contagions [2, 4, 10].

Researchers have identified, through diffusion simulations on networks, many significant differences between the behaviors of simple and complex contagions [2–4, 17]. Among these are increases in times required to reach a cascade (defined in Section 2.1) for the same (random) graph, the decrease in resilience to random failures in scale-free networks, and the significant decrease in the performance of a heuristic to block diffusion, as one moves from simple to complex contagions. See the above references and [18] for a detailed discussion of the differences.

Enhancing the diffusion of complex contagions in social networks is important in applications such as product marketing and others where widespread information propagation is required [6, 27]. In this paper, we carry out an empirical study of the diffusion process for complex contagions with a focus on understanding the role played by the network structure. Our study involves three synthetic networks (an Erdős-Rényi random graph and two scale-free networks constructed using different methods) as well as two practical social networks (namely, epinions and slashdot).

Our primary contributions are as follows. We present theoretical bounds that relate a dynamic property of a network (namely, the maximum number of nodes to which the contagion spreads) to an efficiently computable static parameter (namely, the size of a specific subnetwork) of the corresponding network. These bounds allow us to develop a heuristic for choosing a subset of nodes where the diffusion process can start so as to spread to a large part of the network. We empirically evaluate this heuristic on synthetic as well as practical social networks. Our results show that graphs within the same class (here, scale-free) can exhibit significantly different behaviors depending on the construction method. We also provide detailed empirical results to compare and contrast behaviors of various networks under the two classes of contagions.

The paper is organized as follows. Section 2 describes the diffusion model and related work. Section 3 provides a summary of results. Theoretical results are presented in Section 4. Section 5 describes the networks used in experiments. Study parameters and simulation results are contained in Sections 6 and 7 respectively. Section 8 provides some directions for future work.
2 Dynamical System Model and Related Work

2.1 System Model and Terminology

Our model for studying diffusion in social networks is called a synchronous dynamical system (SyDS) [19]. In this model, each node of a social network represents an individual and each undirected edge \{u, v\} indicates that the contagion may propagate from u to v or from v to u. Each node has a state value which may be 0 indicating that the node is unaffected (i.e., the contagion has not propagated to that node) or 1 indicating that the node has been affected (i.e., the contagion has propagated to that node). Once a node’s state changes to 1, it remains in that state for ever. We refer to this model as a ratcheted SyDS. Other terms such as “progressive” systems have also been used to describe this behavior [15].

State changes at the nodes of a social network are determined by local transition functions. For a node \( v \), let \( f_v \) denote the local transition function at \( v \). The inputs to \( f_v \) are the states of \( v \) and those of the neighbors of \( v \). In our work, each function \( f_v \) is assumed to be a \( t \)-threshold function for some integer \( t \geq 0 \). A precise specification of function \( f_v \) is as follows.

(a) If the state of \( v \) is 1, then the value of \( f_v \) is 1, regardless of the values of the other inputs.
(b) If the state of \( v \) is 0, then the value of \( f_v \) is 1 when \( t \) or more of the inputs to \( f_v \) have the value 1; otherwise, the value of \( f_v \) is 0.

At each time step, all the nodes compute values of their local transition functions synchronously and update their states to the computed values. It should be noted further that, in our model, the local transition functions are deterministic. We are currently studying extensions to probabilistic state transition functions. We use the term \( t \)-threshold system to mean a SyDS in which the local transition function at each node is the \( t \)-threshold function. (The threshold \( t \) is the same for all the nodes.)

At any time, the configuration of a SyDS is a vector \((s_1, s_2, \ldots, s_n)\), where \( n \) is the number of nodes in the social network and \( s_i \) is the state of node \( v_i \), \( 1 \leq i \leq n \). A fixed point is a configuration that does not change; that is, once a SyDS reaches a fixed point, it remains in that configuration for ever. In Figure 1, which shows a 2-threshold SyDS, a fixed point is reached at the end of time 2, with \( v_5 \) transitioning at time 1 and \( v_4 \) transitioning at time 2. In the context of contagions, a fixed point marks the end of the diffusion process. It is known that every ratcheted SyDS where the state values are from \( \{0, 1\} \) reaches a fixed point [19].

We now define some additional terms that are used in the remainder of this paper. For any SyDS, nodes which are in state 1 at time 0 are called seed nodes. Given a SyDS and a particular set of seed nodes, we say that a node \( v \) of the system is reachable if the state of \( v \) is 1 at the end of the diffusion process (i.e., when the system reaches a fixed point). The spread size is the number of
nodes in state 1 (i.e., the number of affected nodes) at the end of the diffusion process. Informally, a cascade occurs when the spread size is a large fraction of all possible nodes; that is, most of the nodes that can be affected are indeed affected. For empirical comparisons, we use a more technical definition of the concept of cascade; this definition will be given in Section 7.

2.2 Related Work

The problem of fostering diffusion for simple contagions was first posed in [6] and they used heuristics to develop solutions. Another efficient approach for the problem was developed in [27]. Numerous works followed based on submodular functions, addressing $\rho$-approximation algorithms to NP-hard generalizations of the problem [12,13,15]. These references also addressed the problem for complex contagions. A further generalization was given in [25]. Work on networks possessing communities [9], and on random graphs for complex contagions [30] has also been reported in the literature. All of these works consider a ratcheted model of behavior. Maximizing influence in the voter model, where nodes can change state back-and-forth between 0 and 1, has been examined [8]. When the threshold for each node is expressed as a fraction of the degree, references [7,24] discuss a necessary and sufficient condition for cascades to occur. Our approach for fostering diffusion of complex contagions relies on identifying subgraphs (called $t$-cores) in which each node has degree at least $t$ for some positive integer $t$. In [14], the concept of cores is effectively used to select seed nodes to enhance the diffusion of simple contagions. However, we could find no study on complex contagions that examines ways to foster diffusion using a core-based approach.

A framework for diffusive models is provided in [1]. That work and [5] address applications to query processing and information retrieval. An example of such a query is the following: given a topic and a semantic network, how can the network be traversed to return highly relevant documents containing information on the topic? Both works describe different classes of constraints
on the diffusion process; in particular, threshold functions are considered an ac-
tivation constraint. The spread of purchasing influence using online sales and
recommendations data collected over a 2-year period have been analyzed [20].
Among other results, they provide a model showing a power law distribution for
the probability of a particular number of product recommendations for different
product classes. The model enables the computation of the probability of a
recommendation cascade based on the data.

3 Summary of Results

As mentioned earlier, a primary goal of our work is to understand the role played
by network structure in fostering diffusion of complex contagions. Towards that
goal, we have developed some simple theoretical results to guide our empirical
study. Our main results are summarized below. Implications of these and ad-
ditional results are also provided in Sections 4 and 7. (In going through the
following summary, the reader should bear in mind that a $t$-core of a graph $G$ is
a subgraph in which each node has degree at least $t$.)

(1) For any $t$-threshold system ($t \geq 1$), we show that the size of the largest
connected component of the maximal $t$-core (defined in Section 4) of the
underlying social network is an upper bound on the maximum spread size
when any seed set of size $t$ is chosen from a $t$-core. We also observe that there
are networks for which this bound is achieved. Since the maximal $t$-core of a
graph can be efficiently computed, this result provides a useful heuristic to
choose seed sets for fostering diffusion of complex contagions.

(2) Our experimental results show that the upper bound on the spread size
mentioned above provides a very good estimate for four of the five networks
evaluated. Hence, a knowledge of the $t$-core size (from a static analysis) can
significantly reduce the number of simulations required to establish the max-
imum achievable spread sizes (a dynamic quantity) for different networks.

(3) When seeds are chosen from a $t$-core, the maximum spread size depends on
the network and the threshold $t$. However, for the five networks tested, the
maximum spread size does not depend on either the number of seed nodes
(as long as it is at least $t$) or on the specific $t$-core chosen for seeding.

(4) When different seed sets (all chosen from appropriate cores) are tried, each
seed set either leads to a large or a very small spread size. Consequently,
there is essentially only one of two possible outcomes, so that average spread
sizes across multiple diffusion instances do not provide a true picture of the
network dynamics.

(5) When different seed sets from $k$-cores are used, the frequency of cascades (i.e.,
proportion of seed sets that cause cascades) depends on $k$. In particular, the
frequency of cascades increases with $k$. The change in cascade frequency in
going from 1-core to 20-core can be a factor of 5 or more. Furthermore, this
effect is often nonlinear in $k$ for the graphs evaluated. This result points out
that it is possible to realize a significant increase in the frequency of cascades
by choosing seed nodes from a 5-core rather than a 1-core.
Graphs within the same class, but constructed using different procedures, can exhibit vastly different diffusion behavior in terms of cascade size and frequency of cascades. This behavior for complex contagions contrasts with that for simple contagions, for as long as the network has a sufficiently large connected component, other aspects of the network structure do not play a significant role in fostering simple contagion diffusion.

4 Theoretical Bounds on Spread Size

Recall that the spread size in a social network is the total number of nodes whose state is 1 at the end of the diffusion process. In this section, we develop bounds on the maximum spread size (a dynamic property of the network) using a static and easily computable parameter of the network. We assume that the given SyDS is a \(t\)-threshold system; that is, each local transition function is the \(t\)-threshold function. We begin with a definition that extends one in [28].

**Definition 1.** Let \(G(V, E)\) be an undirected graph and let \(t \geq 0\) be an integer. A \(t\)-core of \(G\) is an induced subgraph \(G'(V', E')\) of \(G\) such that each node in \(G'\) has a degree of at least \(t\). A \(t\)-core \(G'(V', E')\) of \(G\) is maximal if there is no node \(v\) in \(V - V'\) such that the subgraph of \(G\) induced on \(V' \cup \{v\}\) is also a \(t\)-core.

Even if \(G\) is connected, the maximal \(t\)-core may not be connected. For a graph \(G\) and integer \(t \geq 0\), we use \(L_G(t)\) to denote the number of nodes in the largest connected component of the maximal \(t\)-core of \(G\). As will be seen from the results of this section, the parameter \(L_G(t)\) can be used to bound the maximum spread size in \(t\)-threshold systems.

We now present a result that bounds the maximum spread size in a \(t\)-threshold system in terms of the size of the largest connected component of its maximal \(t\)-core. In stating this result, we assume that the seed set is of size \(t\). If the seed set has less than \(t\) nodes, obviously, there will be no further diffusion in a \(t\)-threshold system.

**Theorem 1.** Let \(G\) be the underlying graph of a \(t\)-threshold SyDS, where \(t \geq 1\). Let \(G'(V', E')\) denote the maximal \(t\)-core of \(G\). For any seed set \(I \subseteq V'\) with \(|I| = t\), the maximum spread size is bounded by \(L_G(t)\), the size of the largest connected component of \(G'\). Moreover, this bound is achievable.

A proof of the above theorem and further discussion when \(I\) has more than \(t\) nodes appear in [18]. In Section 7, we empirically evaluate the above bound for realistic and synthetic social networks.

5 Networks Chosen for the Empirical Study

Five networks are studied. Table 1 summarizes the networks and selected parameters. The first two are realistic social networks, while the last three are...
synthetic networks generated as part of this work. The realistic networks were taken from the literature [21,26]. These networks are directed. However, we converted them to undirected graphs because the undirected graph model is the appropriate one for many social processes. The Erdős-Rényi graph is \( \text{er} \) and the two scale free networks are \( \text{sfpa} \) and \( \text{sffastgen} \). The rationale for our choices of realistic networks and construction parameters of synthetic networks, as well as constructions methods, are given in [18]. The degree distributions of the graphs are depicted in Figure 2(a). The \( k \)-core distributions for the networks are provided in Figure 2(b).

<table>
<thead>
<tr>
<th>Network</th>
<th>Number of Nodes</th>
<th>Number of Edges</th>
<th>Average Degree</th>
<th>Average Clustering Coefficient</th>
<th>Number of Connected Components</th>
<th>Size of Largest Component</th>
</tr>
</thead>
<tbody>
<tr>
<td>epinions</td>
<td>75879</td>
<td>405740</td>
<td>10.7</td>
<td>0.138</td>
<td>2</td>
<td>75877</td>
</tr>
<tr>
<td>slashdot</td>
<td>77360</td>
<td>469180</td>
<td>12.1</td>
<td>0.0555</td>
<td>1</td>
<td>77360</td>
</tr>
<tr>
<td>( \text{er} )</td>
<td>99998</td>
<td>500172</td>
<td>10.0</td>
<td>0.000106</td>
<td>1</td>
<td>99998</td>
</tr>
<tr>
<td>sfpa</td>
<td>100000</td>
<td>499959</td>
<td>10.0</td>
<td>0.000994</td>
<td>1</td>
<td>100000</td>
</tr>
<tr>
<td>sffastgen</td>
<td>100007</td>
<td>568277</td>
<td>11.4</td>
<td>0.0127</td>
<td>1032</td>
<td>97921</td>
</tr>
</tbody>
</table>

Table 1. Two realistic and three synthetic networks and selected characteristics.

Fig. 2. (a) Degree distributions and (b) \( k \)-core sizes for the networks, through the 20-core.

6 Experimental Parameters and Procedures

6.1 Simulations

An iteration is the execution of a diffusion process on a network, with a specified set of seed nodes and state transition functions. A simulation is a set
of iterations. In this work, each simulation is composed of 100 iterations, with each iteration using a different set of seed nodes. Seed sets were computed off-line prior to simulations; this enables us to use the same seed sets for different $t$-threshold functions in comparing results.

Each seed set is chosen as follows. A node $j$ is randomly selected from the specified $k$-core to be the first anchor. Its neighbors are randomly made seeds, one at a time, until the required number of $n_s$ seeds is obtained. If $j$’s neighbors are exhausted and there are less than $n_s$ seeds, then one of the current seeds is randomly made the new anchor and the process repeats. In this manner, the set of seed nodes induces a connected subgraph; see [18] for further details.

6.2 Experimental Parameters

Table 2 contains the parameters and their values used in this experimental study. Simulations were run with all combinations of the parameters in this table. For each $n_s$, $k$-core, and network combination, 100 different sets of seed nodes, each of size $n_s$, were selected from a $k$-core to assess the effect of different seed sets on diffusion. Hence, each simulation used seed nodes from a particular $k$-core across all iterations. For each simulation, all nodes have the same threshold $t$.

<table>
<thead>
<tr>
<th>Graphs</th>
<th>$k$-Cores</th>
<th>Thresholds, $t$</th>
<th>Numbers of Seeds, $n_s$</th>
</tr>
</thead>
<tbody>
<tr>
<td>epinions, slashdot, er, sfpa, sffastgen</td>
<td>1, 5, 20</td>
<td>2, 3, 5</td>
<td>2, 3, 5, 10, 20, 50, 100, 200</td>
</tr>
</tbody>
</table>

Table 2. Parameters and values of parametric study. Simulations were conducted for all combination of the above parameter values. Each simulation is comprised of 100 iterations, corresponding to 100 sets of seed nodes $I$, all of the same size $|I| = n_s$.

7 Simulation Results

We introduce the formal notion of cascade, which is used throughout this section. A **cascade** is said to occur when the spread size is at least $\gamma n_r$, where $0 < \gamma \leq 1$ is a specified value and $n_r$ is the sum of the number of nodes in the maximal $t$-core for $t$-threshold diffusion, plus the number of seed nodes that are not in the $t$-core. This choice of $n_r$ is based on the results of Section 4. We take $\gamma = 0.9$ in defining a cascade, for reasons made clear below. Since the number of seed nodes used in our experiments is much less than the sizes $n_t$ of the maximal $t$-cores examined ($t \leq 5$), we set $n_r = n_t$.

We have run simulations up through $n_s = 200$. However, as the results presented here do not change beyond $n_s = 50$ (in the sense that curves for $n_s = 50, 100$, and 200 are the same), our plots only show data through $n_s = 50$.  

27
Erdős-Rényi Graph For the \text{er} graph, it was found that virtually no diffusion occurred in any of the simulations. This suggests that an average degree of 10 in an Erdős-Rényi random graph with 100,000 nodes is insufficient to connect the network to an extent that permits appreciable propagation of complex contagions. An informal explanation for this behavior is that for the \text{er} graph, the probability of an unaffected node becoming affected in the first time step of a diffusion simulation is small (roughly $2 \times 10^{-4}$); see [18] for details. Consequently, most of the results in subsequent sections do not include the \text{er} graph.

Baseline Diffusion Results Recall that the spread size is the cumulative number of nodes affected at the end of an iteration, which is a fixed point. Our experimental procedure computes the spread sizes for each of the 100 iterations to determine the number of iterations that cascade. Note that in addition to seeding the $t$-core—a hypothesis of Theorem 1—we also seed $t'$-cores for other values of $t'$.

Figure 3 shows results for \text{sffastgen} for seeding from the 5-core with $t = 2$. Each curve corresponds to a different number $n_s$ of seeds. For each $n_s$, the spread sizes for the 100 iterations are arranged in increasing order and plotted. Note that the largest spread size—the cascade size—of 51% of all graph nodes is independent of $n_s$. This graph shows that 76% of all iterations attain maximum spread for $n_s = 3$ because the blue curve transitions at 24% of iterations. As $n_s$ increases, the fraction of iterations reaching a large spread size increases.

We stated in relation to Figure 3 that the cascade size is constant. To be precise, the cascade values may change ever so slightly, at most by about 1.5% of nodes, from the smallest to largest cascade sizes. For clarity of presentation, we ignore such minor effects.
Predicting the Maximum Size of a Cascade  In Figure 3 we used the term “cascade” for large spread sizes without justifying the fact that they meet our definition of a cascade, provided at the beginning of this section. We do this now as part of the process of experimentally evaluating the upper bound on the maximum spread size given by Theorem 1.

In this study, the maximum size of a cascade is independent of the k-core that is seeded. This is to be expected since seeding even the 1-core does not preclude a large spread size for, say \( t = 3 \); examples to this effect can be readily generated. However, the k-core is important in determining the maximum possible number of affected nodes, for a given threshold, as described in Section 4. We now evaluate experimentally the bound established in Theorem 1.

Figure 4 shows the average fraction of nodes affected (i.e., the average relative spread size) for each of the five networks as a function of threshold value. It is constructed as follows. Let \( n_c \) be the average spread size from all iterations for a particular network and \( t \) threshold diffusion process. We saw in the previous figure that this maximum spread value is independent of \( n_s \) for sffastgen; this is true for all networks of this study. The maximum spread size is also independent of the k-core that is seeded [18]. We do not include spread sizes that are less than 5% of the network nodes in the computation of \( n_c \), for we are evaluating the upper bound on the spread size. We divide \( n_c \) by the size \( n_t \) of the appropriate t-core for the respective network and plot these points in Figure 4. The horizontal line with ordinate value 1.0 (shown using black circles and black dashed line), indicates perfect agreement predicted by Theorem 1 between spread size and core size. Data for the five networks are solid curves. It is seen that, indeed, the t-core size provides an upper bound on the spread size, since \( n_c/n_t \leq 1 \). Values of \( n_c/n_t \) vary between 0.93 and 1.0 for different values of \( t \) for four of the networks, with all but one value greater than 0.95. Furthermore, sfpa demonstrates that the upper bound is achievable, per Theorem 1. However, the bound is very poor for er, since very little diffusion occurs in that network. Thus, the upper bound of Theorem 1 is tight for four of the five large networks considered in our study.

Based on our definition of cascade—that the spread size be \( \geq \gamma n_r \)—we now see that \( \gamma = 0.9 \) covers all spread sizes that are significantly greater than zero, because for four networks, \( n_c/n_r = n_c/n_t \geq 0.93 \). This justifies our use of the term cascade for large spread sizes throughout this paper.

We illustrate the usefulness of Theorem 1 by re-examining Figure 3. We noted that the cascade size was 51% for \( t = 2 \). From Figure 2(b), we obtain that the maximum spread size for \( t = 2 \) for sffastgen is 52%—a very tight bound to the experimentally determined value of 51%. What happens for \( t = 3 \)? Figure 2(b) also predicts the maximum spread size for \( t = 3 \) to be 37%. This, too, is a very tight upper bound to the experimentally determined value of 36%. So the k-cores alone tell us that the cascade size will decrease from 52% to 37% as \( t \) increases from 2 to 3. That is, without running a single simulation, a comparison of the upper bounds on cascade sizes (a parameter of the system dynamics) can be made from a one-time k-core distribution determination for a network (a static parameter). Clearly, in practice, we want to run some simulations,
because er shows us that the predicted bound may be weak, but the utility of the static analysis is clear. Moreover, in this instance, the comparison was made for one graph, but comparisons can also be made between graphs. We will provide additional examples in subsequent sections. Moreover, this type of result is not confined to sffastgen; Figure 4 shows that four of the five graphs used in this study will give these types of results.

**Effect of Network Generation Process For Scale-Free Networks**

Two of the synthetic networks are scale-free: sfpa and sffastgen. The diffusion behavior is different between these networks, as seen in Figure 5: sfpa generates larger cascade sizes and sffastgen generates a greater frequency of cascades. For sfpa in Figure 5(a), the percentage of nodes affected in cascades is 100%; for sffastgen in Figure 5(b), the size of cascades is only 51% of network nodes. sffastgen has smaller cascade sizes even though it has a larger average clustering coefficient. (In this study, only the sfpa network exhibited cascades that include 100% of nodes.) On the other hand, the sffastgen network generates more cascades. For example, for $n_s = 3$, only 4% of iterations cascade for sfpa, while for sffastgen this number is 64%. These comparisons hold across all $t = 2, 3, 5$, $k = 1, 5$, and $n_s = 2$ to 200.

![Figure 5. Example plots showing the spread size for (a) sfpa and (b) sffastgen with $t = 2$, $k = 1$, and different values of $n_s$.](image)

These behaviors are consequences of the network generation methods and network structures. For sfpa, since each node has degree $\geq 5$, we have from Section 4 that $L_G(5)$ is the number of nodes in $G$; that is, the 5-core is the entire graph. For $t \leq 5$, by Theorem 1, all nodes can become affected for these sets of simulation conditions with sfpa. (This is not the case for $t > 5$.) So, cascades, if they occur, can permeate through the network, and indeed they do. Also, 47% of the nodes in sffastgen are of degree 1, and these cannot be reached by complex contagions. The counterpoint is that for each value of degree $\geq 50$, sffastgen has more nodes than sfpa, and sffastgen’s degree distribution extends to higher degree values. Therefore, its hubs are more effective in
spreading contagion, resulting in a greater fraction of cascades. The point is this: the network generation process, even within one class of graphs, can have a major effect on the diffusion behavior of complex contagions. For simple contagions, these effects would not exist: all nodes of a connected component would comprise a cascade and cascades would occur in 100% of iterations when a graph has a dominant connected component.

Theorem 1 is also useful in comparing the spread sizes of \textit{sfpa} and \textit{sffastgen}. In Figure 5 we noted that for $t = 2, 3, \text{ and } 5$, the sizes of cascades for \textit{sfpa} were greater than those for \textit{sffastgen}. We can confirm these experimental observations using Theorem 1 along with Figure 2(b), which is based on the sizes of $t$-cores. Thus, our experimental results point out that cascades close in size to the upper bound value can be achieved, as stated in the theorem. However, the theorem and Figure 2(b) tell us unambiguously that for $t \geq 6$, the ranking of networks will change: the size of a cascade for \textit{sffastgen} will now be at least as great as that for \textit{sfpa} because the size of the maximal 6-core for \textit{sfpa} is zero and hence the maximum cascade size is zero.

As an aside, Figures 3 and 5(b), which seed two different $k$-cores but otherwise have the same input parameters, demonstrate that the $k$-core seeded does not affect the cascade size as claimed above.

Effect of $k$-Core Seeded and Threshold on Number and Sizes of Cascades

Figure 6 shows effects of the seeding method on the fraction of iterations that cascade. The abscissa in both plots represents the value of $k$ such that seed nodes are selected from a $k$-core. Figure 6(a) shows results for the \textit{epinions} network. The solid green curve shows that for $t = 3$ and $n_s = 5$, if seeds are selected from the 1-core, then 47% of the 100 iterations cascade. If seeds are selected from the 5-core and 20-core, the fraction of cascades climbs to 81% and then 99% of iterations. Also, for $t = 2$ and $n_s = 2$ (orange dashed line), increasing the $k$-core seeded from the 1-core to the 20-core increases the fraction of cascades from 0.18 to 0.94, an increase by a factor of five. These results show that as the value of $k$ increases, the greater the chance of a cascade, and that this effect is most often nonlinear. Hence, even a relatively small increase in core, from the 1-core, can make a big difference in the fraction of cascades. Both \textit{slashdot} and \textit{sffastgen}—the only other graphs whose $k$-cores change to permit these comparisons—also exhibit these same magnitudes in sensitivity of probability of cascade to seeded core. Simply seeding from a higher core, therefore, increases the propensity for cascades; hence, in an average sense, nodes from a larger core are more influential.

From a practical standpoint, if one wants to maximize the chances for a cascade, one should seed the highest core possible. Assuming that it takes more resources to seed higher $k$-cores (people who are more well-connected may be more expensive to reach and convince), one can still reap benefits by seeding higher cores by, say, seeding the 5-core, rather than the 1-core, without going to the possibly greater expense of seeding the 20-core. If identifying higher-core
groups of agents is costly, an alternative is to seed more nodes, because increasing \( n_s \) also increases the fraction of cascades in Figure 6(a).

![Figure 6](image_url)

**Fig. 6.** Effect of \( k \)-core seeded on the fraction of iterations that cascade: (a) epinions network and (b) four networks and \( n_s = 10 \).

Selected results across networks are provided in Figure 6(b) for \( n_s = 10 \). First, not all effects of \( k \)-core are nonlinear (cf. sffastgen and \( t = 5 \)), but the great majority are, at least for the conditions of this study. Second, some curves may be concave up; cf. slashdot and \( t = 5 \). Third, sfpa allows far fewer cascades than other networks for the same conditions. (The curves are horizontal for sfpa because the 1-core and 5-core are the same for this network, so the same results are used for both.) Fourth, the ranking of networks, in descending order of number of cascades, is not always the same across conditions. For example, with \( t = 3 \) (the solid curves), epinions and sffastgen show comparable behavior, with epinions having fewer cascades for the 1-core, while sfpa has slightly fewer cascades for the 5-core. The slashdot network has fewer cascades than epinions and sffastgen for all k-cores when \( t = 3 \). However, for \( t = 5 \), the comparisons change dramatically. The epinions network now permits far more cascades than the other two networks, possibly because of its greater average clustering coefficient. Also, slashdot now permits more cascades than sffastgen. Developing a rigorous justification of these behaviors is a challenging research issue.

8 Future Work

In subsequent work, we plan to evaluate these diffusion-fostering methods against more straight-forward methods, such as seeding high degree nodes. We will evaluate heterogeneous thresholds among nodes and probabilistic diffusion models. The latter includes probabilistic state transition when \( t \) neighbors are affected, probabilistic state transitions that permit small probabilities of state change even when the number of neighbors affected is less than \( t \), and probabilistic pair-wise
interactions between nodes. We also intend to explore the bound on maximum possible spread size to understand the conditions under which the bound is tight and identify situations where the bound can be improved.

Acknowledgments. We thank the reviewers from the Analysis of Complex Networks (ACNE) Workshop of ECML PKDD 2010 for their inputs. We also thank our external collaborators and members of the Network Dynamics and Simulation Science Laboratory (NDSSL) for their suggestions and comments. This work has been partially supported by NSF Nets Grant CNS-0626964, NSF HSD Grant SES-0729441, NIH MIDAS project 2U01GM076094-7, NSF PetaApps Grant OCI-0904844, DTRA R&D Grant HDTRA1-0901-0017, DTRA CNIMS Grant HDTRA1-07-C-0113, NSF NETS CNS-0831633, DHS 4112-31805, NSF CNS-0845700 and DOE DE-SC003957.

References


Cross-Domain Literature Mining through Outlier Document and Bridging Concept Detection

Nada Lavrač¹,², Borut Sluban³ and Matjaž Juršič¹

¹ Jožef Stefan Institute, Ljubljana, Slovenia
² University of Nova Gorica, Nova Gorica, Slovenia
³ International Postgraduate School Jožef Stefan Institute, Ljubljana, Slovenia
{nada.lavrac, borut.sluban, matjaz.jursic}@ijs.si

Abstract. In literature-based creative knowledge discovery the goal is to identify interesting terms or concepts which relate different domains. This paper reveals that a majority of these concepts can be found in outlier documents which are not in the mainstream domain literature which describes matters related to a common understanding of the domain of investigation. To find bridging concepts in cross-domain literature mining, this paper proposes to identify two seemingly unrelated domains of interest, detect outlier documents in the two domains which have joint intersecting terms, and identify bridging terms indicating yet unexplored links between the two domains. We have developed two novel outlier detection algorithms and evaluated their knowledge discovery potential on the classical migraine-magnesium cross-domain discovery problem.

Keywords: text mining, creative knowledge discovery, outlier detection

1 Introduction

This paper investigates the role of outliers in the area of literature mining, contributing a novel approach to text mining (Feldman and Sangers 2007). It explores the utility of outliers in a non-standard text mining task of cross-domain link discovery. The motivation for our focus on outlier documents in the literature has grounds in Mednick’s associative creativity theory (1962) and in the literature on domain-crossing associations, called bisociations, introduced in Koestler’s book The act of creation (Koestler 1964). Mednick defines creative thinking as the capacity of generating new combinations of distinct associative elements (e.g. words). He explains how thinking about the concepts that are not strictly related to the elements under investigation inspires unforeseen useful connections between the elements. On the other hand, according to Koestler, a bisociation is a result of creative processes of the mind when making completely new associations between concepts from domains that are usually considered separate. Consequently, bisociations may considerably improve creative discoveries. Through the history of science, this mechanism has been a crucial element of progressive insights and paradigm shifts. Nevertheless, no comprehensive literature mining methodology has yet been developed on this basis.

This work provides evidence that outlier detection methods can contribute to literature-based cross-domain scientific discovery based on the notion of bisociation. The presented approach to creative knowledge discovery from text documents is based on exploring interesting terms in outlier documents, used to detect implicit relationships across different domains of interest. It is an instance of closed discovery¹

¹ In contrast to closed discovery, open discovery leads the creative knowledge discovery process from a given starting domain towards a yet unknown second domain which at the end of this process turns out to be connected with the first one.
defined by Weeber et al. (2001), where two domains of interest, A and C, are identified by the expert prior to starting the knowledge discovery process.

Weeber et al. (2001) have followed the work of creative literature-based discovery in medical domains by Swanson (1986) who designed the ABC model approach that investigates whether an agent A is connected with a phenomenon C by discovering complementary structures via interconnecting phenomena B. Two literatures are complementary if one discusses the relations between A and B, while a disparate literature investigates the relations between B and C. If combining these relations suggests a previously unknown meaningful relation between A and C, this can be viewed as a new piece of knowledge that may contribute to a better understanding of phenomenon C.

In a closed discovery process, both C and A are known and the goal is to search for bridging concepts (terms) b in B in order to support the validation of the hypothesis about the hypothesized connection between A and C. Smalheiser and Swanson (1998) developed an online system ARROWSMITH, which takes as input two sets of titles from disjoint domains A and C and lists b-terms that are common to literature A and C; the resulting bridging terms (b-terms) are used to generate novel scientific hypotheses (see Figure 1). As stated by Swanson et al. (2006), the major focus in literature-based discovery has been on the closed discovery process, where both A and C are specified in advance.

![Figure 1. Closed discovery process as defined by Weeber et al. (2001).](image)

The method proposed in this paper focuses on finding the bridging terms in outlier documents of two given domains A and C. Outlier documents provide a focus for detecting bridging terms, enabling the exploration of potentially interesting bisociative links between the two domains. These links may be indicative of new insights and as such, they may contribute to bisociative knowledge discovery. To verify the proposed approach, our methodology has been applied to a challenging medical problem: discovery of novel links from the migraine–magnesium literature explored by Weeber et al. (2001).

This paper is organized as follows. Section 2 presents and relates two creative knowledge discovery frameworks: the Koestler's bisociative link discovery and the Swanson’s ABC model of closed discovery in literature mining (Swanson 1986). It also relates our work to a previous literature mining methodology RaJoLink (Petrić et al. 2009). In Section 3 we show how a set of documents is transformed into a network format and explain a specific network representation - so-called bisociative information network, shortly BisoNet – which is used in this work. Section 4 sets the stage for detecting outlier documents as a first step for guiding the knowledge discovery process.

---

2 Uppercase letter symbols A, B and C are used to represent sets of terms, and lowercase symbols a, b and c to represent single terms.
discovery process. We present two types of methods for outlier detection, classification noise filters (Brodley and Friedl, 1999) and an approach based on cartification (Goethals and Vreeken, 2010), which we have adapted to outlier document detection in a cross-domain knowledge discovery setting. Section 5 presents the heuristics used in selecting the most promising bridging concepts (linking terms) in outlier documents. Evaluation of the proposed approach on the migraine-magnesium domain pair, explored originally in Swanson’s research, is provided. In Section 6 we conclude by a discussion and directions for further work.

2 Relating Koestler’s creativity model to comparable cross-domain knowledge discovery approaches

This section presents some background on the mechanism of bisociation. Bisociative reasoning is at the heart of creative, accidental discovery (e.g., serendipity), and is focused on finding unexpected links by crossing domains. Scientific discovery requires creative thinking to connect seemingly unrelated information, for example, by using metaphors or analogies between concepts from different domains. These modes of thinking allow the mixing of conceptual categories or domains, which are normally separated. One of the functional bases for these modes is the idea of bisociation, coined by Artur Koestler (1964):

“The pattern . . . is the perceiving of a situation or idea, L, in two self-consistent but habitually incompatible frames of reference, M1 and M2. The event L, in which the two intersect, is made to vibrate simultaneously on two different wavelengths, as it were. While this unusual situation lasts, L is not merely linked to one associative context but bisociated with two.”

Koestler found bisociation to be the basis for human creativity in seemingly diverse human endeavors, such as humor, science, and arts. The concept of bisociation is illustrated in Figure 2.

We explore a specific pattern of bisociation: links between concepts which belong to different domains. The creative act is to find links which lead ‘out-of-the-plane’ in Koestler’s terms, i.e. links which cross two or more different domains. More precisely, we claim that two concepts are bisociated if:

- there is no direct, obvious evidence linking them,
- one has to cross domains to find the link, and
- this new link provides some novel insight into the problem domain.

![Figure 2. Koestler’s schema of bisociative discovery (Koestler 1964).](image)

Although Koestler’s insight into creative knowledge discovery is almost half a century old, no comprehensive literature mining methodology has yet been developed on this basis.

In this paper we explore an approach to bisociative cross-domain link discovery, based on outlier documents, used to detect implicit relationships across different pre-defined domains of expertise. It can be seen that - in terms of the Swanson’s ABC model used in literature mining - this is an approach to closed knowledge discovery, where two domains of interest, A and C, are identified by the expert in advance. In
terms of the Koestler’s model, the two domains, $A$ and $C$, correspond to the two habitually incompatible frames of reference, $M1$ and $M2$. Moreover, the linking $b$-terms that are common to literature $A$ and $C$, explored by Smalheiser and Swanson (1998), clearly correspond to Koestler’s notion of a situation or idea, $L$, which is not merely linked to one associative domain but bisociated with two domains $M1$ and $M2$. These observations are summarized in Table 1.

<table>
<thead>
<tr>
<th>Koestler’s model</th>
<th>Swanson’s model</th>
</tr>
</thead>
<tbody>
<tr>
<td>bisociative link discovery process</td>
<td>closed discovery process</td>
</tr>
<tr>
<td>frames of reference (domains) $M1$ and $M2$</td>
<td>domains of interest $A$ and $C$</td>
</tr>
<tr>
<td>bisociative cross-context link $L$ in $M1 \cap M2$</td>
<td>bridging term $b$-term in $\text{terms}(A) \cap \text{terms}(C)$</td>
</tr>
</tbody>
</table>

Table 1. Unifying Koestler’s and Swanson’s models of creative knowledge discovery.

In a closely related approach, rarity of terms as means for knowledge discovery has been explored in the RaJoLink system (Petrič et al. 2009; Urbančič et al. 2007), which can be used to find interesting scientific articles in the PubMed database, to compute different statistics, and to analyze the articles with the aim to discover new knowledge. The RaJoLink method involves three principal steps, Ra, Jo and Link, which have been named after the key elements of each step: Rare terms, Joint terms and Linking terms, respectively. In the Ra step, interesting rare terms in literature about the phenomenon $C$ under investigation are identified. In the Jo step, all available articles about the selected rare terms are inspected and interesting joint terms that appear in the intersection of the literatures about rare terms are identified as the candidates for $A$. This results in a candidate hypothesis that $C$ is connected with $A$. In order to provide explanation for hypotheses generated in the Jo step, in the Link step the method searches for $b$-terms, linking the literature on joint term $a$ from $A$ and the literature on term $c$ from $C$. Note that steps Ra and Jo implement the open discovery, while step Link corresponds to the closed discovery process, searching for $b$-terms when $A$ and $C$ are already known (as illustrated in Figure 1).

3 Documents representation and network creation

In bisociative knowledge discovery, as explored in this work, selected knowledge representation formalism is a so-called bisociative information network, called BisoNet. A BisoNet and related bisociation concepts are defined as follows:

- A BisoNet is a large graph, where nodes are concepts and edges are probabilistic relations. Unlike semantic nets and ontologies, it carries little semantics and to a large extend encodes just circumstantial evidence that concepts are somehow related through edges with some probability.
- A context (frame of reference) is represented by a BisoNet subgraph. An assignment of different subgraphs to different contexts is subjective to the user (domain expert). This is to support a move from her/his ‘normal’ context to a habitually incompatible context, i.e. a creative jump ‘out-of-the-frame’.
- A bisociation is an implicit link (to be discovered) between nodes or subgraphs from different contexts.

In order to create a BisoNet from text documents, a data preprocessing phase is required. In our work, data preprocessing consists of

- creating a bag-of-words (BoW) representation of documents (described below in this section), and
- document outlier detection (described in Section 4).
Data preprocessing is followed by the creation of a BisoNet from BoW documents (described below in this section) and the detection of bridging terms (described in Section 5).

3.1 Creating BoW vector representation of documents in data preprocessing

Data preprocessing aims at building a vector representations (bag of words, BoW) of the documents. It consists of the following steps:

- Stopword removal. This standard text mining step is not used, due to extremely short documents in the given medical document sets consisting of titles/abstracts only (we used a domain specific stopword list later in the processing of documents when constructing a BisoNet, but not in the data preprocessing step).
- Word stemming/lemmatization. LemmaGen lemmatiser for English is used (Juršič et al. 2007).
- Determining term length. Maximal term length (N-gram length) parameter is set to three (i.e. the maximal length of bridging terms; e.g.: calcium channel blocker is a term consisting of three words).
- Determining minimal word frequency. The minimum frequency parameter determines the frequency of occurrence of the words in a text which are to be included in the vocabulary. This parameter is set to two.

3.2 Creating a BisoNet graph from BoW vectors

In order to create a BisoNet network, network nodes and edges need to be defined. Since the purpose of network creation is to simplify the exploration and discovery of bridging terms (b-terms) used in linking two domains (contexts) A and C, only the potentially relevant terms are defined as network nodes. A BisoNet is therefore constructed in the following way:

- Nodes are all the terms (including b-terms but without stopword terms) that have their bisociation potential (BP) value greater than a predefined node weight threshold. The BP value of a term is defined as a product of the term’s TF-IDF value in the centroid vector of domain A times the term’s TF-IDF value in the centroid vector of domain C: \( bp(Term) = \text{tfidf}(Term, A) \times \text{tfidf}(Term, C) \), where A and C are defined by their centroid vectors.
- Links between nodes are calculated using the bisociativity index (BI), defined by Segond and Borgelt (2009)\(^3\). Only those links are created that have the BI value greater than a predefined link weight threshold.

A BisoNet graph can be dynamically displayed and explored by the expert to find bridging links and b-terms. Besides standard functions for interactively navigating through the graph (moving, zooming, unzooming, ...) the exploration step contains also two filtering controls. With them, the expert controls the size of the graph displayed and implicitly sets the ratio of importance between the b-potential and the bisociativity index measures. These two controls are implemented as follows:

- Node weights (BP) threshold slider – only those nodes are displayed that have the weight higher than the weight set on this control (see Figure 8 in the appendix).
- Link weights (BI) threshold (slider) – only those links are displayed that have the weight higher than the weight set on this control (see Figure 8 in the appendix).

\(^3\) Bisociativity index was defined by Segond and Borgelt (2009) as follows:

\[
BI(t^A, t^C) = \sum_{i=0}^{k} \left( \sqrt{tf^A_i \cdot tf^C_i} \cdot \left( 1 - \frac{\text{tan}^{-1}(tf^A_i) - \text{tan}^{-1}(tf^C_i))}{\text{tan}^{-1}(1)} \right) \right) \cdot tf^A_i \cdot tf^C_i.
\]
4 Detecting outlier documents

Creative thinking requires focusing on problems from new perspectives with the ability to bridge the gap between different domains. Such relations between distinct domains can be revealed through the bridging concepts. Since this may lead to the generation of many possible ideas, the innovative composition of hypotheses as well as support for facilitated exploration of alternatives are needed for creative cross-domain knowledge discovery.

Based on this assumption, we have experimented with two methods for closed knowledge discovery. In both, outliers are used as heuristic guidance to speed-up the search for bridging concepts between different domains of expertise. The intuition behind this research is that outlying documents in the domain literature have a higher probability to provide observations that may lead to the discovery of bridging concepts. In this way the outliers can be employed for finding new interesting relations among the dispersed literatures of different domains.

In summary, in our research we aim at finding cross-domain links between concepts from two disparate literature sources $A$ and $C$, based on exploring outliers in the articles of the two domains. Our method assumes that by exploring outlier documents it is be easier to discover linking $b$-terms (bridging concepts) that establish previously unknown links between literature $A$ and literature $C$.

4.1 Experimental domain: Literature on migraine and magnesium

To evaluate the proposed literature outlier detection and $b$-term extraction approach we applied the proposed method to explore an important application domain. We investigated the ability of the proposed approach to detect previously unknown relationships between migraine and magnesium. To this end, we replicated the early Swanson’s migraine-magnesium experiment that represents a gold standard for literature-based discovery. Therefore, the evaluation procedure used in this experiment differs from the original Swanson’s method and the RaJoLink method in that a human expert was not involved.

Similar to Swanson in his original study of the migraine literature (Swanson 1988) we used titles as input to our closed discovery process. We performed the experiments on a subset of PubMed titles of articles that were published before 1988 (i.e., before Swanson’s literature-based discovery of the migraine-magnesium relation) and were retrieved with the PubMed search engine. As a result we got migraine and magnesium titles of PubMed articles, excluding those which would include both terms migraine and magnesium in the same title. These article titles were analyzed by our outlier detection methods in the closed discovery setting. Unlike Swanson we focused on those titles from migraine and magnesium literature identified by our methods as interesting outliers in either the migraine or magnesium literature.

Magnesium deficiency has been shown in several studies to cause migraine headaches (e.g. Swanson 1990; Thomas et al. 1992; Thomas et al. 2000; Demirkaya et al. 2001; Trauninger et al. 2002). In the literature-based closed discovery process Swanson managed to find more than 60 pairs of articles connecting the migraine domain with the magnesium deficiency via several bridging concepts. His closer inspection of the literature about migraine and the literature about magnesium showed that 11 pairs of documents, when put together, provided confirmation of a hypothesis that magnesium deficiency may cause migraine headaches (Swanson 1990).

4.2 Classification noise filters for outlier detection

The novelty of our work is to use noise detection approaches to finding outlier documents containing cross-domain links (bridging $b$-terms) between different domains. We used the Swanson’s migraine-magnesium dataset (obtained by searching
PubMed for documents including these two keywords, after preprocessing resulting in 7,930 articles. We first searched for a set of outlier documents with a classification noise filtering approach (Brodley and Friedl, 1999), implemented and adapted for this purpose. Classification noise filtering is based on the idea of using a classifier as a tool for detecting noisy and outlier instances in data. This outlier detection method works in a 10-fold cross-validation manner, where repeatedly nine folds are used for training the classifier and on the complementary fold the misclassified instances are denoted as noise/outliers. We implemented three classification noise detection algorithms, using three different classifiers: Naïve Bayes (abbreviated: Bayes), Support Vector Machine (SVM) and Random Forest (RF).

To measure the relevance of the detected outlier documents in terms of their potential for containing domain bridging terms, we inspected 20 terms known as bridging terms appearing in the given preprocessed migraine-magnesium domain pair. We compared their relative frequencies in the detected outlier sets to their relative frequencies in the whole dataset. The three classification filters implemented with different classifiers, Bayes, SVM and RF, found 765, 416, and 763 outliers, respectively. To our satisfaction, in these three sets of outlying documents, 17 (Bayes), 13 (SVM) and 17 (RF) of the 20 bridging terms are present. The relative frequencies of these terms in the sets of detected outlying documents are presented in Figure 3. The results show that nearly all of the bridging terms present in the set of outlying documents have a significantly higher relative frequency in this set compared to their frequency in the whole dataset.

Figure 3. Comparison of relative frequencies of bridging terms (t1, ... , t20) in the following document sets: the entire dataset and in the sets of outlier documents detected by three different outlier detection methods.

Figure 4. Comparison of relative frequencies of bridging terms (t1, ... , t20) in the following document sets: the sets of outlier documents detected by three different outlier detection methods and the randomly sampled document sets of same size.
We have also randomly sampled subsets of documents from the migraine-magnesium dataset, which had the same size as the sets of outlier documents found by the three classification filters. For each of the outlier set sizes (765, 416 and 763) ten document sets were randomly sampled. A comparison of the relative frequencies of bridging terms in the outlier document sets and the average relative frequencies of bridging terms in the randomly sampled sets of the same size is presented in Figure 4.

The results in Figures 3 and 4 show that in the sets of outlier documents the relative frequencies of bridging term are not only higher than in the whole dataset (all documents), but are also higher than in randomly sampled document sets of same size. These results are thereby confirming our assumption, that exploring outliers can serve as a significant speed-up for searching for terms that bridge different domains.

4.3 Cartification-based outlier detection

The cartification approach, as proposed by Goethals and Vreeken (2010), can also be used for outlier detection. Inspired by frequent itemsets used in association rule learners, the cartification approach creates for every instance in the dataset a transaction, or cart, in which the k-nearest neighbors of that instance are stored. The resulting collection of carts can then be used to mine frequent itemsets; that is, sets of points that are frequently seen together in some dimensions. Instances that appear in very few carts can be considered as outliers, as these instances do not have much in common with other instances.

We have adapted the algorithm to enable outlier detection as follows. In our setting we compute carts for set A and set C. The size of carts is set to the square root of the size of the set. Thereby sets of outlier documents of document sets A (migraine) and C (magnesium), denoted as $O(A)$ and $O(C)$ respectively, are obtained. The main novelty of our method is that these document sets, $O(A)$ and $O(C)$, are inspected further in the joined context of domains A and C. For each of these outlier documents, $k$-nearest neighbors are computed and if the majority of its $k$ nearest neighbors belong to the other domain than the given outlier is considered a good candidate for containing domain bridging terms.

After the cartification process each document belongs to at least one or more carts and the documents appearing in only few carts are considered as outliers. In our experimental setting we used two cutoff values for determining outliers. The first one denoted $col0$ considers as outliers only document appearing in the least number of carts (i.e., documents with the smallest cart frequency), and the second $col1$ considers the documents with cart frequencies among the 0.1% lowest within the cart frequency range.

Figure 5 shows the relative frequencies of domain bridging terms ($b$-terms) obtained with cartification-based outlier detection (bridging documents detection) compared to relative frequencies of bridging terms in the whole dataset.

**Figure 5.** Comparison of relative frequencies of bridging terms ($t_1,...,t_20$) in the following document sets: the entire dataset and in the sets of outlier documents detected by cartification with different cutoff values.
The results show that all the bridging term that were present in the sets of outlier documents detected by the cartification approach with cut-off value 0 (k89co0) have higher relative frequencies in the selected outlier document sets than in the entire dataset. The approach with the higher cutoff value (k89co1) yielded document sets which contained more bridging terms, but resulted in lower relative frequencies of the terms found by k89co0. However, the majority of the additionally found bridging terms in the outlier document set obtained by k89co1 have a higher relative frequency than in the entire dataset.

To support the relevance of the results of higher relative frequencies of bridging terms in outlier document sets, we have again randomly sampled sets of documents corresponding to sizes 204 and 1303 as obtained by the cartification-based outlier detection approaches k89co0 and k89co1, respectively. The comparison of relative frequencies of bridging terms in outlier document sets and in randomly sampled document set of same size can be found in Figure 6.

![Figure 6. Comparison of relative frequencies of bridging terms (t1,..., t20) in the following document sets: the sets of outlier documents detected by cartification with different cutoff values and the randomly sampled document sets of same size.](image)

### 4.4 Summarizing the results

Srinivasan and Libbus (2004) developed an algorithm for bridging concept identification that is claimed to require the least amount of manual work in comparison with other studies. However, it still needs substantial time and human effort for collecting evidence relevant to the hypothesized connections. In comparison, one of the advantages of our approach is that the domain expert needs to be involved only in exploring the potential $b$-terms in outlier documents, instead of exploring all the most frequent potential $b$-terms in all the documents.

This result indicates that the proposed method supports the creative discovery of bridging concepts between different domains, which are usually not considered together. Outlying documents are thus able to indicate if there exists a term (i.e. a bridging concept), which can be used to form bisociations. Thus the identification of outliers can improve the closed discovery process by substantially speeding-up the search of $b$-terms.

Experimental results on the migraine-magnesium domain clearly show that instead of searching for $b$-terms in all the documents of the two domains, the expert can focus on a substantially reduced number of outlier documents. Both classification noise filtering approach and the cartification approach proved effective in finding outlier documents containing nearly all the bridging terms, thus suggesting that the expert’s time needed in searching for $b$-terms can be substantially reduced.
5 Detecting b-terms in outlier documents

This section focuses on identifying bridging concepts (b-terms) between two domains. The first domain consists of titles of articles about migraine and the second about magnesium. The b-terms identified by Swanson (1988) are used as a gold standard in the evaluation. They were verified by the expert and provide a new discovery in the field.

5.1 Heuristics for b-term detection

Our method of b-term detection (either in the entire document set, or preferably just in the set of outlier documents detected by outlier detection algorithms, presented in Section 4) is the following.

1. Employ text mining to pre-process the texts and encode them in the bag-of-words representation;
2. Calculate the heuristics which favor b-terms over other terms;
3. Sort domain terms by the best heuristic measure and present the top terms (hopefully representing b-terms) to the expert during interactive exploration of the two domains.

The search for the most promising heuristic is based on two phases:

- Training – we proposed over 40 heuristics, from very simple term-frequency statistics to very elaborate combined measures. We evaluated their quality on the migraine-magnesium gold standard and selected the best one, called the b-potential heuristic.
- Testing - we independently evaluated the b-potential measure on a second dataset, autism-calcineurin documents, to confirm its domain independence and quality of b-term identification.

We experimentally confirmed that the proposed b-potential measure is the best heuristic for b-term detection and is able to retrieve b-terms approximately 7 times faster as compared to a random approach. The b-potential measure is calculated as a multiplication of the term’s tf-idf weights in the centroids of the two domains.

Figure 7 shows ROC curves evaluating different heuristics in terms of the quality of b-term retrieval, given lists of terms ranked by the individual heuristics on the migraine-magnesium training dataset. The ROC curve of Figure 7 is constructed by drawing a vertical line when a term is indeed a b-term, and a horizontal line when a term is not a b-term. Therefore, the y-axis shows the number of b-terms and the x-axis shows the number of non-b-terms. The figure presents a selection of best heuristics, where a comparison of the best b-potential heuristic (leftmost) with the random (rightmost) is indicative.

5.2 Summarizing the results

The experimental results indicate that when during interactive exploration the expert is presented with a list of relevant terms sorted by b-potential, the probability that the list contains a b-term is 7 times higher compared to a random list (equivalently, this list is expected to contain 7 times more b-terms than a random list).

---

4 ROC curves are well known from machine learning and data mining literature as means for classifier performance evaluation, as well as evaluation of rankings.
6 Conclusions

Current literature-based approaches depend strictly on simple, associative information search. Commonly, a literature-based association is computed using measures of similarity or co-occurrence. Because of their ‘hard-wired’ underlying criteria of co-occurrence or similarity, association-based methods often fail to discover relevant information which is not related in obvious associative ways. Especially information related across separate domains is hard to identify with the conventional associative approach. In such cases the domain-crossing connections, called bisociations, can help generate creative and innovative discoveries.

Swanson (1986) and Weeber et al. (2001) have investigated means for finding novel interesting connections between disparate research findings which can be extracted from the published literature. They have shown that the analysis of implicit associations hidden in scientific literature can guide the hypotheses formulation and lead to the discovery of new knowledge.

The method presented in this paper has the potential for bisociative link discovery as it allows switching between domains. Similar to the Swanson’s closed discovery approach (Swanson 1986), the search for bridging concepts consists of looking for \( b \)-terms that can be found in separate sets of records: in some literature \( A \) as well as in some other literature \( C \). However, our focus is on outliers from the two sets of records and their neighboring documents which leads to substantial search reduction.

The presented methodology for literature mining implementing closed cross-domain link discovery consists of the following steps:

- Identify two seemingly unrelated domains of interest.
- Detect outlier documents in the two domains with joint intersecting terms.
- Identify bridging terms to find previously unexplored links between the two domains (one of the goals of this step is selection of a subset of outlier documents, containing most of the bridging terms).

We have shown that outlier documents yield useful information in the closed discovery setting, where connections have to be found between literatures \( A \) and \( C \). In fact, such visual analysis can present previously unseen relations, which provide new knowledge. This is an important aspect and significant contribution of our method to literature-based discovery research.

In our experiments, the proposed method has succeeded to identify useful outlier documents, containing bridging concepts (\( b \)-terms) between the literature on migraine and magnesium. Moreover, by further exploring the outlier document, the effort needed for finding linking concepts (\( b \)-terms) is substantially reduced, as it requires to explore a much smaller set of potential \( b \)-terms.
Acknowledgements

The work presented in this paper is supported by the European Commission under the 7th Framework Programme FP7-ICT-2007-C FET-Open project BISON-211898, and by the Slovenian Research Agency grants Knowledge Technologies (P2-0103), and Semantic SoKD (J2-2353).

References


Appendix

Figure 8. Bisociation visualizer application: expert can display a network of terms and interactively browse through it. One also has the option to filter out terms and/or links with weights lower than some specified value (the slider controls at the top of the application). Both images show the application on the same data, with the only difference that the top image shows higher threshold setting than the bottom one. It is clear that increasing threshold manifests in more nodes and links displayed.
Constraint-Based Mining of Sets of Cliques
Sharing Vertex Properties

Pierre-Nicolas Mougel\textsuperscript{1,2}, Marc Plantevit\textsuperscript{1,3}, Christophe Rigotti\textsuperscript{1,2}, Olivier Gandrillon\textsuperscript{4}, and Jean-François Boulicaut\textsuperscript{1,2}

\textsuperscript{1} Université de Lyon, CNRS, INRIA
\textsuperscript{2} INSA-Lyon, LIRIS Combining, UMR5205, F-69621, France
\texttt{firstname.lastname@liris.cnrs.fr}
\textsuperscript{3} Université Lyon 1, LIRIS Combining, UMR5205, F-69622, France
\textsuperscript{4} Université Lyon 1, Centre de Génétique Moléculaire et Cellulaire, UMR5534, F-69622, Villeurbanne, France
\texttt{lastname@cgmc.univ-lyon1.fr}

Abstract. We consider data mining methods on large graphs where a set of labels is associated to each vertex. A typical example of such graphs is a social network of collaborating researchers where additional information represent the main publication targets (preferred conferences or journals) for each author. We investigate the extraction of sets of dense subgraphs such that the vertices in all subgraphs of a set share a large enough set of labels. As a first step, we consider here the special case of dense subgraphs that are cliques. We proposed a method to compute all \textit{maximal homogeneous clique sets} that satisfy user-defined constraints on the number of separated cliques, on the size of the cliques, and on the number of labels shared by all the vertices. The empirical validation illustrates the scalability of our approach and it provides experimental feedback on two real datasets, more precisely an annotated social network derived from the DBLP database and an enriched biological network of protein-protein interactions. In both cases, we discuss the relevancy of extracted patterns thanks to available domain knowledge.

Keywords: graph mining, network analysis, pattern discovery, constraint-based mining

1 Introduction

Many data can be represented by means of graphs where vertices represent entities and edges represent relationships between entities. For instance, graphs provide a natural representation of important real life networks such as biological networks and social networks. In the last years, such network data became increasingly available and network/graph mining indeed turned out to be one of the most studied and challenging tasks for the data mining community. Many researchers developed approaches to mine graphs in two different and complementary ways. On the one hand, some methods focus on macroscopic graph properties (e.g., degree distribution, diameter) \cite{9} or graph partitioning \cite{12}. On
the other hand, many proposals concern the extraction of more sophisticated properties within a pattern discovery setting. In particular, local pattern mining in graphs has received a lot of attention, leading to the introduction of new problems (like support counting in case of non-relational graphs [7, 8]) and resulting in new algorithms to mine collections of graphs [38, 25, 15, 36], single graphs [18, 27], and time-evolving graphs [6, 3, 32].

Most of the existing methods work on graph data only. However, as mentioned, for instance in [21], more informative graphs are often given that can be represented as graphs with feature vectors associated to each vertex. In [21], the authors take advantage of the complementarity of the information carried by edges and features, and extract dense subgraphs such that the vertices in one subgraph share a large enough set of features. In this paper, we consider that datasets denote graphs where a set of labels is associated to each vertex. Such a set can represent a set of Boolean property values and can also be used to encode a discrete feature vector (when feature domains are transformed to be disjoint). Over such graphs, we investigate the extraction of sets of dense subgraphs such that the vertices in all subgraphs of a set share a large enough set of labels. As a first step, we consider here the special case of dense subgraph known as a clique.

Following a pattern discovery setting within the constraint-based mining framework, we introduce the problem of extracting maximal homogeneous clique sets which are sets of cliques that satisfy constraints on the number of separated cliques, on the clique sizes and on the number of labels shared by all the vertices. We propose an efficient algorithm to extract maximal homogeneous clique sets in a complete way. It should be noticed that enumerating all cliques is a time consuming operation (due to the large number of cliques), and that potentially enumerating all the clique sets is obviously even more difficult. Let us now motivate our research thanks to applications from system biology and social network analysis.

Example 1 (Mining Protein-Protein Interaction Graphs). Protein-protein interaction databases contain a large number of interactions. These interactions can be modeled as a protein-protein interaction graph where vertices are proteins and two vertices are connected if the two proteins are known to interact. In biology, a functional module is a cluster of proteins that interact together in a specific cellular process. Thus, cliques often model functional modules. Protein-protein interaction graph can be enriched by gene expression data since proteins are products of genes. In our framework, situations where genes are overexpressed can be considered as labels. [15, 21, 35] consider the same kind of cross fertilization and enable to discover meaningful patterns. Maximal homogeneous clique sets, can help the expert to discover different functional modules that share a large enough set of biological situations. This kind of knowledge is highly valuable since it gives rise to links between these modules. These links can provide evidences of cooperative or competitive actions of groups of genes overexpressed in the same biological situations.

Example 2 (Social Network Analysis). One of the most studied task in social network analysis is the discovery of communities [37]. A community is, for instance,
a group of people who share some interests and are connected by strong social interaction. Communities can be modeled as dense subgraphs (e.g., cliques) in which vertices share a large enough set of properties. Extracting maximal homogeneous clique sets, can help to identify sets of communities that share the same interests. For instance, in a co-authorship network (e.g., DBLP\(^5\)), a maximal homogeneous clique set can describe groups of co-authors that are active on the same topics. It can be used, for instance, to design program committees and selection processes.

We provide an experimental feedback in this two contexts (protein interaction network and co-authorship network) on real datasets, showing that using constraints enable to focus on small collections of meaningful patterns. Furthermore, we show that the extraction process is scalable: it can be performed on large graphs with hundreds of thousands of vertices.

The rest of the paper is organized as follows. Section 2 introduces the definitions and the mining methods. Our experiments are reported in Section 3. Related work is discussed in Section 4. Section 5 briefly concludes.

2 Maximal homogeneous clique set

In this section, we define the patterns of interest and introduce a correct method to compute them.

2.1 Pattern definition

**Definition 1.** (Dataset)

Let \( \mathcal{L} \) be a set of labels, a dataset is a pair \( \langle G, f \rangle \), where \( G = \langle V, E \rangle \) is a simple undirected graph (vertices \( V \) and edges \( E \)), and \( f \) is a total function \( f : V \to 2^\mathcal{L} \) associating a set of labels to each vertex.

A set \( C \) of vertices is called a **clique** in \( G \) if the subgraph of \( G \) induced by \( C \) is complete. The collection of all cliques in \( G \) is denoted \( \mathcal{C}_G \).

**Definition 2.** (Homogeneous Clique Set) Let \( \alpha, \beta, \) and \( \kappa \) be three strictly positive integers, a **Homogeneous Clique Set** (HCS) in dataset \( \langle G, f \rangle \) is a collection \( M \) of cliques \( \{C_1, \ldots, C_n\} \subseteq \mathcal{C}_G \) such that the three following constraints \( C^\text{lab}_{\alpha}, C^\text{clique}_{\kappa, \beta} \) and \( C^\text{sep} \) are satisfied:

- \( C^\text{lab}_{\alpha} : |\bigcap_{C \in M} (\bigcap_{v \in C} f(v))| \geq \alpha \), i.e., the vertices share at least \( \alpha \) labels;
- \( C^\text{clique}_{\kappa, \beta} : M \) contains at least \( \kappa \) cliques of size at least \( \beta \);
- \( C^\text{sep} : \) for all \( C, C' \) in \( M \), with \( C \neq C' \), we have \( C \cup C' \notin \mathcal{C}_G \), i.e., the cliques in \( M \) are separated.

\(^5\) http://www.informatik.uni-trier.de/~ley/db/
It should be noticed that the cliques in a HCS are not required to be maximal cliques in $G$ because this would be too restrictive. Furthermore, the $C_{\text{sep}}$ constraint is needed to avoid a large clique of $G$ to be split and counted as several cliques in a HCS.

For a dataset, the collection of HCS is likely to be large, so we focus on the maximal ones, that are in some sense the most specific.

**Definition 3.** (Maximal Homogeneous Clique Set and partial order $\preceq$) A Maximal Homogeneous Clique Set (MHCS) is a HCS which is maximal w.r.t. the partial order $\preceq$ defined as follows. Given $M_1$ and $M_2$ two HCS, $M_1 \preceq M_2$ if and only if for all $C_1 \in M_1$ there exists $C_2 \in M_2$ such that $C_1 \subseteq C_2$.

In general, antisymmetry does not hold for $\preceq$ in any collection of sets, but in the special case of a collection of HCS $\preceq$ is a partial order as stated by the following theorem.

**Theorem 1.** The relation $\preceq$ is a partial order in a collection of HCS.

*Proof.** The relation is trivially reflexive and transitive. To show antisymmetry, consider $M_1$ and $M_2$ two HCS such that $M_1 \preceq M_2$ and $M_2 \preceq M_1$. Suppose that $M_1 \neq M_2$, then there exists $C$ in $M_1$ that is different from all sets in $M_2$. And since $M_1 \preceq M_2$, there exists $C'$ in $M_2$ such that $C \subseteq C'$. As, $M_2 \preceq M_1$, there exists $C''$ in $M_1$ such that $C' \subseteq C''$. So, $C \subseteq C''$, and as elements of a HCS must be separated, this cannot hold. □

The maximal homogeneous clique set problem consists in finding all the MHCS in a given dataset.

### 2.2 Finding all MHCS

Let $\text{vertices}(M)$ be the set of vertices appearing in a collection $M$ of sets of vertices, $\text{vertices}(M) = \bigcup_{C \in M} C$. The following theorem states that if $M$ is a MHCS, then knowing $\text{vertices}(M)$ is sufficient to determine $M$.

**Theorem 2.** Given $M$ a MHCS, then $M$ is the collection of all maximal cliques in the subgraph $G_M$ of $G$ induced by $\text{vertices}(M)$.

*Proof (sketch).* Let $S$ be the collection of all maximal cliques in $G_M$ and suppose that $M \neq S$. Since $S$ contains all maximal cliques, and $M$ contains cliques that must be separated, then $S \not\subseteq M$. So, there exists $D \in S$ such that $D \not\in M$.

If $D$ is a superset of a clique $C$ in $M$, since cliques in $M$ are separated, replacing $C$ by $D$ in $M$ leads to a pattern that satisfies $C_{\text{sep}}$ (and also $C_{\text{lab}}^{\alpha}, C_{\text{clique}}^{\kappa,\beta}$), and thus $M$ is not a maximal HCS.

If $D$ is not a superset of a clique in $M$, since $D$ is a maximal clique in $G_M$ then all cliques in $M \cup \{D\}$ are separated. So $M \cup \{D\}$ satisfies $C_{\text{sep}}$ (and also $C_{\text{lab}}^{\alpha}, C_{\text{clique}}^{\kappa,\beta}$), and again $M$ is not a maximal HCS. □
For a dataset \((G, f)\), \(f\) can be encoded as a binary relation \(R \subseteq V \times L\), defined as \(xRy \iff y \in f(x)\), and relating each vertex to its labels. The MHCS are related to the so-called closed sets over \(R\) as states by the next theorem.

Let us consider the mappings \(g\) and \(h\), defined as follows, \(g : 2^V \rightarrow 2^L, g(X) = \{y \in L | \forall x \in X, xRy\}\) and \(h : 2^L \rightarrow 2^V, h(Y) = \{x \in V | \forall y \in Y, xRy\}\). These mappings define a Galois connection between \(2^V\) and \(2^L\) (e.g., see [39, 28]), a set of vertices \(V \subseteq V\) (resp. of labels \(L \subseteq L\)) is said closed in \(R\) if \(V = h(g(V))\) (resp. \(L = g(h(L))\)), and when restricted to closed sets, the mappings \(g\) and \(h\) are anti-isomorphisms (i.e., bijections that assign to \(A, B\) s.t. \(A \subseteq B\), the images \(A', B'\) s.t. \(B' \subseteq A'\)).

**Theorem 3.** Given \(M\) a MHCS, then \(\text{vertices}(M)\) is closed in \(R\).

**Proof (sketch).** Let \(V = \text{vertices}(M), L = g(V), V' = h(L),\) and suppose that \(V\) is not closed in \(R\), then \(V \subseteq V'\). Let \(M'\) be the collection of all maximal cliques in the subgraph of \(G\) induced by \(V\). This collection satisfies \(\mathbb{C}^{\text{sep}}\). Since \(M\) is a MHCS, then by Theorem 2 \(M\) is the collection of all maximal cliques in the subgraph of \(G\) induced by \(V\), and thus \(M \preceq M'\), and \(M'\) satisfies \(\mathbb{C}^{\text{clique}}\) because \(M\) satisfies it. Since \(V' = h(L)\), the vertices in \(V'\) share at least as many labels as the vertices in \(V\). As \(M\) satisfies \(\mathbb{C}^{\text{lab}}\), \(M'\) satisfies it also. So \(M\) is not a maximal HCS.

Let \(\text{maxCliques}(G, V)\) be the collection of maximal cliques in the subgraph of \(G\) induced by \(V\). Theorems 2 and 3 lead to the two following correct ways to find all MHCS:

- Find the closed sets of vertices \(V\) in \(R\), such that \(\text{maxCliques}(G, V)\) satisfies \(\mathbb{C}^{\text{lab}}, \mathbb{C}^{\text{clique}}\) and \(\mathbb{C}^{\text{sep}}\). Among them retain the maximal ones. For each of this maximal set \(V\), output \(\text{maxCliques}(G, V)\).
- Or alternatively, using the Galois connection, find the minimal closed sets of labels \(L\) in \(R\), such that \(\text{maxCliques}(G, h(L))\) satisfies \(\mathbb{C}^{\text{lab}}, \mathbb{C}^{\text{clique}}\) and \(\mathbb{C}^{\text{sep}}\), and then for these minimal \(L\), output \(\text{maxCliques}(G, h(L))\).

**Constraint properties**

**Definition 4.** (Set of vertices and set of labels satisfying the constraints) A set of vertices \(V\) (resp. set of labels \(L\)), \(V\) (resp. \(L\)) satisfies the constraints \(\mathbb{C}^{\text{lab}}, \mathbb{C}^{\text{clique}}\) or \(\mathbb{C}^{\text{sep}}\) if and only if the collection \(\text{maxCliques}(G, V)\) (resp. the collection \(\text{maxCliques}(G, h(L))\)) satisfies the same constraints.

Let us consider these constraints and their properties of monotonicity (i.e., if \(A\) satisfies a constraint then all supersets of \(A\) satisfy it) and anti-monotonicity (i.e., if \(A\) satisfies a constraint then each subset of \(A\) satisfies it). The following properties are straightforward.

- \(\mathbb{C}^{\text{lab}}\) is monotonic (resp. anti-monotonic) w.r.t. the sets of labels (resp. the sets of vertices);
\( C_{\kappa,\beta} \land C_{\text{sep}} \) is anti-monotonic (resp. monotonic) w.r.t. the sets of labels (resp. the sets of vertices).

Additionally, the conjunction \( C_{\kappa,\beta} \land C_{\text{sep}} \) can be expressed in a relaxed form by means of the constraint \( C_{\text{vert}} \) defined as follows: \( C_{\text{vert}} \) holds for \( M \) if \(|\text{vertices}(M)| \geq \beta + \kappa - 1\) (i.e., in order to contain at least \( \kappa \) separated cliques of size at least \( \beta \), \( M \) must be built over at least \( \beta + \kappa - 1 \) vertices).

The anti-monotonicity (resp. monotonicity) w.r.t. the sets of labels (resp. the sets of vertices) of \( C_{\text{vert}} \) are obvious as well. Since checking \( C_{\text{vert}} \) is easy and do not require to know the maximal cliques, we can use it first, and then check \( C_{\kappa,\beta} \land C_{\text{sep}} \) only when \( C_{\text{vert}} \) is satisfied.

### Algorithm and implementation

So, to extract the MHCS, we can choose between enumerating sets of vertices or sets of labels, while pushing the constraints when applicable. In the targeted application domains, as shown in Section 3, \( \mathcal{L} \) is likely to be smaller than \( \mathcal{V} \), so in the rest of the paper we consider the enumeration of sets of labels in \( \mathcal{R} \). Notice that, in the context of extracting the so-called formal concepts in gene expression data, enumerating the sets of labels (representing in this context set of biological situations) has been shown to be an interesting approach [26, 31], and turns out to be feasible in practice, even when approaches based on the enumeration of sets of genes was not. However, an experimental comparison remains to be done, to confirm the interest of this strategy for MHCS.

To compute the MHCS through the enumeration of closed sets of labels, we can easily reuse most depth-first or levelwise closed set mining algorithms, handling \( C_{\text{vert}} \) as a standard support constraint (anti-monotonic), \( C_{\alpha}^{\text{lab}} \) as a monotonic constraints, and \( C_{\kappa,\beta} \land C_{\text{sep}} \) as another anti-monotonic constraint. In our current implementation we use an algorithm similar to Closet [29]. The conjunction \( C_{\kappa,\beta} \land C_{\text{sep}} \) is pushed partially, by pruning the current branch if it is not satisfied. We do not push the monotonic \( C_{\alpha}^{\text{lab}} \) constraint, that is only checked in a passive way, but it could be used actively in future developments using, for instance, the ExAnte data reduction technique [5].

Since we want only the minimal closed sets of labels satisfying the constraints (the ones corresponding to the maximal HCS), when a closed set \( L \) that fulfills the requirements is found then the current branch is pruned, and \( L \) and \( \text{maxClique}(G, h(L)) \) are stored. And finally, when the exploration terminates, to guarantee the minimally of the closed sets, a test is performed over these sets in a post-processing step, and \( \text{maxClique}(G, h(L)) \) is output for each minimal set \( L \).

To extract the maximal cliques (function \( \text{maxClique} \)), we implemented the algorithm of [34] that have an optimal worst-case time complexity.

### 3 Experiments

We evaluate our algorithm on two real-world datasets. On the first one, a social network dataset, we report a qualitative study and quantitative results. On
the second one, a biological dataset, we present a detailed qualitative interpretation of a relevant pattern. In these experiments, we found original patterns with strong added value which would not be found with usual local pattern mining tasks. Quantitative experiments show that the algorithm scales up on large datasets. All experiments were performed on a PC running GNU/Linux with a 3 GHz Core 2 Duo CPU and 8 GB of main memory installed (no more than 700 MB used by the extraction process). Our experimental setting aims at answering the following questions: Does our pattern definition provide new kind of knowledge and make sense? Can domain experts easily exploit the collection of patterns? Does our approach scale well on large datasets?

3.1 Mining DBLP data

The social network dataset is built from the public DBLP database. This database contains a rather exhaustive bibliographic information from most of computer science conferences and journals. It has been extensively used as an experimental dataset by many researchers. Notice also that scientific collaboration networks have similar properties than social networks [24]. Our dataset is built using all conferences since 2000 included. A vertex represents an author and an edge between two authors means that they have coauthored at least two papers. A vertex is labeled with the conference names in which the corresponding author has published (e.g., ICDM, KDD).

In the first experiment, for an author, in its label list, we retain only the conferences in which she/he has published at least twice (to discard one-shot involvements). Authors with an empty remaining label list are removed. The resulting graph has 117,526 vertices (authors), 467,691 edges (coauthor relationship of at least two papers) and there are 3,257 different labels (conferences). In this dataset, we search for MHCSs with at least 3 cliques of 3 vertices and 6 common labels ($\alpha = 6$, $\beta = 3$ and $\kappa = 3$). 80 patterns respect those constraints. Among them, 32 patterns are related to at least one of the following data-mining conferences: ICDM, KDD or SDM. We focus on two of these patterns, presented Figure 3.1. The pattern in Figure 1(a) contains 5 cliques:

\{Jian Pei, Jiawei Han, Ke Wang, Philip S. Yu, Jeffrey Xu Yu, Ke Wang, Philip S. Yu, Christos Faloutsos, Philip S. Yu, Spiros Papadimitriou, Jiawei Han, Philip S. Yu, Wei Wang\}. [Hans-Peter Kriegel]

The first one contains four authors: Jiawei Han, Jian Pei, Philip S. Yu and Ke Wang. It is well known that these prolific authors have been working together over the past decade. Furthermore, we can see that the vertex corresponding to Philip S. Yu has a particular role, looking like a hub for 4 of the cliques. Discovering such local hubs in a subnetwork can be important and useful.

The other pattern, presented Figure 1(b), contains 3 cliques:

\{Jian Pei, Jiawei Han, Philip S. Yu, Jiawei Han, Jiong Yang, Philip S. Yu, Christian Böhm, Hans-Peter Kriegel, Peer Kröger\}

\footnote{We do not set an edge between two authors that have coauthored only one paper, since we think that this cannot be interpreted as a real collaboration.}
The third clique is not connected to the two others, Christian Böhm, Hans-Peter Kriegel and Peer Kröger are all working in the same university located in Germany, whereas the two other groups are formed by people located in North America (working in the same universities at some time). This kind of local structure is particularly interesting since it exhibits groups that are not known to interact, but that share similar interests.

Now, we report the CPU times and the numbers of patterns obtained in a series of quantitative experiments with different settings for \( \alpha \), \( \beta \) and \( \kappa \). In these experiments, in order to get a larger dataset, for an author we retain all conferences where she/he has published, and no author is removed. The resulting dataset contains 479,067 vertices, 773,613 edges and 3,607 different labels.

Regarding main memory usage, we consider the maximal memory usage during each extraction. This maximal value never exceed 700 MB, and is about 657 MB on average over all extractions with a standard deviation of 7 MB. Concern-
ing time performances, Figures 2(a) and 2(b) show that the extractions can be made in reasonable time, even when constraints are weakly selective. The worst case is obtained for $\alpha = 6$, $\beta = 3$, and $\kappa = 2$ and requires less than 17 minutes. As the algorithm starts by enumerating all closed sets having at least $\alpha$ labels, time performances depend mostly on $\alpha$ for small values of $\alpha$. When $\alpha$ increases, pruning the search space thanks to $C_{vert}$ and $C_{clique}$ is more effective and thus $\beta$ and $\kappa$ have more impact on time performances. Regarding the number of output patterns, Figures 3(a) and 3(b) show that it shrinks fast when parameter values increase. For $\alpha > 1$, when $\beta$ increases by two, the number of patterns decreases by at least one order of magnitude.

![Graphs showing the evolution of the number of patterns](image.png)

**Fig. 3.** Evolution of the number of patterns (log scale) w.r.t. $\alpha$, $\beta$ and $\kappa$

### 3.2 Mining biological data

In this experiment, we built a dataset by using two databases: STRING [14] and SQUAT [19]. STRING aggregates data on protein-protein interactions from different sources (i.e., genomic data, co-expression, literature). Genes encode the proteins, so such interaction data can be read as gene-gene interactions (interactions of the proteins that are encoded by the genes). Among these interactions, we only retained interactions with a confidence\(^7\) higher or equal to 400 (default STRING selection threshold). SQUAT is a Boolean (discretization over 0/1) gene expression database containing results from SAGE experiments (the discretization process is explained in [2]). SQUAT was created to support post-genomic data analysis processes for several species, and contains, for thousands of genes, the sets of biological situations (termed libraries) where these genes are overexpressed. In our experiments, we used only Human species genes. SQUAT contains information about only a subset of STRING genes, and thus we removed from STRING the proteins encoded by genes that do not correspond to genes in SQUAT (using HUGO names for the mapping).

So, to sum up, from STRING we obtained a graph where vertices represent genes and edges represent interactions between these genes (interactions

---

\(^7\)This confidence is a measure provided by STRING. Low confidence means that there are not so many evidences that the interaction exists.
between the proteins encoded by these genes), and using SQUAT we associated to each vertex a set of labels representing the set of biological situations (SAGE libraries) where this gene is overexpressed. The resulting dataset contained 4,988 vertices (genes), 70,126 edges (interactions) and 486 different labels (biological situations).

In [13] and [35], the authors argue that combining gene expression and protein interactions leads to promising results. The main interest of our MHCS approach when mining such a combination of information is the following. Since the cliques do not need to be interconnected, it should enable to extract MHCS containing groups of genes that are functionally unrelated except for proteins making some bridges, for example, a transcription factor that activates genes that fall in different functional categories, a pattern that we indeed could extract (see Figure 4). This pattern was extracted using the following parameters: $\alpha = 3$, $\beta = 3$, and $\kappa = 2$.

![STRING interaction graph of the genes forming a MHCS with 2 cliques of 3 genes overexpressed in 3 situations](image.png)

**Fig. 4.** STRING interaction graph of the genes forming a MHCS with 2 cliques of 3 genes overexpressed in 3 situations

First, we have been studying the relevancy of the pattern using the L2L tool [23]. It is immediately apparent that the best p-values was obtained for the “Visual perception” category of the GO biological processes with a highly significant score ($p = 6.17 \cdot 10^{-11}$). This was due to the six following genes products that are related to eye development and vision: PDE6G, PPEF2, ABCA4, RBP3, RDH8, and CRX. One should note that RAXL1 also harbor retina-related functions (see below) that were not detected by L2L. We then investigated the nature of the three libraries (i.e., labels) found within the pattern. It turned out that all three libraries were made from normal retina\(^8\), which is perfectly rele-

---

\(^8\) The three libraries are:

- SAGE
- Retina
- Peripheral
- normal
- Peripheral
vant given the nature of the extracted genes. We observed that the CRX gene behaved as a hub in the pattern, being the more densely connected vertex. We therefore investigated the nature of CRX. Using the hyperlink from SQUAT to Entrez Gene, we could find the following description of the function of CRX: “The protein encoded by this gene is a photoreceptor-specific transcription factor which plays a role in the differentiation of photoreceptor cells”. We then turned to examine the two cliques: (1) CRX, ABCA4, RAXL1 and (2) CRX, GNB1, PDE6G. Both cliques contain, and are associated by, the CRX gene product. This confirms its role as a hub, consistent with its transcriptional factor function. The first clique associates the Retinal-specific ATP-binding cassette transporter (ABCA4) as well as the Retina and anterior neural fold homeobox like 1 (RAXL1). RAXL1 encodes a transcription factor and ABCA4 encodes a membrane protein susceptible to transport retinal. This is therefore a clique centered upon the retinal functions of the proteins it harbors. The second clique associates Guanine nucleotide-binding protein G(I)/G(S)/G(T) subunit beta 1 (GNB1) and the Retinal rod rhodopsin-sensitive cGMP 3',5'-cyclic phosphodiesterase subunit gamma (PDE6G). The first gene encodes a G-protein and the second one encodes the effector molecule in G-protein-mediated phototransduction in vertebrate rods and cones. This clique appears as a quite homogeneous clique involved in signal transduction and under the control of the CRX transcription factor. So the final interpretation is that we have extracted information regarding the overexpression for a photoreceptor-specific transcription factor in retinal cells, together with 12 genes, most of them harboring known function in the retina. This motif is centered upon CRX, which act as a connection between the two cliques, one centered on G-protein-mediated signal transduction, the other unrelated. This demonstrates the relevancy and the actionability of patterns discovered by the MHCS extraction method.

4 Related work

Two kinds of approaches have been proposed to mine graphs whose vertices are described by set of labels. On the one hand, clustering methods were proposed in [12, 13, 35]. On the other hand, some proposals consider a local pattern discovery approach, generally in a constraint-based mining setting. In [21], the authors introduce the problem of cohesive pattern mining in feature vector graphs (each vertex is associated to a feature vector that represents properties of this vertex). Cohesive patterns are subgraphs that satisfy a subspace cohesion constraint (i.e., they share a large enough set of features), a density constraint, and a connectivity constraint. [20] extends cohesive pattern mining to quantitative features. These approaches do not enable to discover patterns that involve several subgraphs. It should be noticed that, in [3], properties are also associated to vertices of a time-evolving graph. The authors define a method to discover rules from

SAGE_Retina normal B_4cRet
SAGE_Retinal Pigment Epithelium normal B_4MacRPE
local graph patterns that characterize the evolution of the graph. However, this approach was not defined to capture several subgraphs within the same pattern.

The problem of mining maximal homogeneous clique sets can be seen as a constraint-based pattern mining task on two data sources (e.g., graph data and transactional data). In this way, our approach is similar to [10] where the authors define a generic framework to extract patterns under a rich set of constraints. They exploit the cross-fertilization of data sources by mining micro-array data for patterns that must also satisfy some constraints on other datasets (e.g., a similarity matrix computed from textual information). However, this approach was not defined for graph data mining.

[15, 11] consider different problems on the same kind of datasets. [15] mines for cross-graph quasi-cliques, and in [11], the authors tackle the problem of redescription mining, that aims at finding subgroups having several descriptions. We think that such approaches are complementary to the one proposed in this paper.

A maximal homogeneous clique set can be seen as a set of patterns (cliques) which satisfy both local constraints and constraints that require to consider several local patterns. In [33], the authors propose the so-called exception rules that combine three local patterns (three different rules). Recently, several generic approaches - pattern teams [17], constraint-based pattern set mining [30], constraint programming for $n$-ary patterns [16] - aim at selecting patterns from the initial set of local patterns to return a smaller and more valuable set according to the context.

Since its introduction, research in pattern mining has aimed at discovering more valuable knowledge nuggets. From the simple frequency constraint, many primitives have been defined and several classes of constraints are now well understood. Pattern domains have become more sophisticated and meaningful. Recently, researchers have considered heterogenous and distributed data combined with domain knowledge to discover implicit relation between concepts from different domains, providing some novel insight into the problem domain [1, 4, 22]. Our work can be seen as being at the frontier between this bisociative knowledge discovery and constraint-based pattern mining.

5 Conclusion

In this paper, we considered the combined mining of a graph and of a binary relation associating sets of labels to the vertices. We proposed to search for patterns called maximal homogeneous clique sets, that are sets of cliques such that all vertices in a pattern shared a large enough set of labels. We described how the selection criteria on these patterns can be used as constraints in a complete extraction method. Finally, we reported experiments, showing that these extractions can be made on real datasets, and can lead to meaningful patterns.

Acknowledgments. This work is partly funded by the French contract ANR-07-MDCO-014 Bingo2 (Knowledge Discovery For and By Inductive Queries).
References

1. BISON: Bisociation networks for creative information discovery (http://www.bisonet.eu/)
38. Yan, X., Han, J.: gSpan: Graph-Based Substructure Pattern Mining. In: Int. Conf. on Data Mining (ICDM). pp. 721–724 (2002)
Heuristic best-first search in separation of interleaved Web sessions

Marko Poženel, Viljan Mahnič, Matjaž Kukar

University of Ljubljana, Faculty of Computer and Information Science,
Tržaška 25, SI-1001 Ljubljana, Slovenia,
marko.pozenel@fri.uni-lj.si, viljan.mahnic@fri.uni-lj.si,
matjaz.kukar@fri.uni-lj.si

Abstract. We describe a heuristic search-based method for interleaved HTTP (Web) session reconstruction building upon first order Markov models. Interleaved session is generated by a user who is concurrently browsing a web site in two or more web sessions (browser windows). In order to assure data quality for subsequent phases in analyzing user’s browsing behavior, such sessions need to be separated in advance. We propose a separating process based on best-first search and trained first order Markov chains. We develop a testing method based on various measures of reconstructed sessions similarity to original ones. We evaluate the developed method on two real world clickstream data sources: a web shop and a university student records information system. Preliminary results show that proposed methods perform well.

1 Introduction

Data about behavior of web site visitors have become one of the most important sources of information in most web-aware companies. They play an important part in daily transactions and important business decisions. It is essential to get reliable data analyses, which require both appropriate methods and data. The quality of the patterns discovered in data analysis depends on the quality of the data on which data mining is performed. The main source of data for the analysis of user behavior represents clickstream data [9]. A sequence of clicks that user makes browsing through a web site is called a clickstream. A user session is represented by one visit of a user to a web site. For better web usage mining results we need reliable sessions. Clickstream data from a normal web-site are noisy, page events are often not explicitly linked to page requests. The pre-processing phase is therefore prone to errors. Although many methods for sessions reconstruction have been devised [1, 11], reliable session reconstruction still remains a challenge.

Really interested and capable users often browse the same web site with multiple browser windows opened. In each web browser they perform actions to complete different tasks. Typically, users switch between browsing tasks so that they work on a single task only for a certain time period. Even if only one user is currently active, concurrent sessions may actually happen, each for
one web browser window (task). In a web server log file all concurrent sessions will be seen as a single long session. We call such sessions interleaved sessions. Such sessions cannot be easily separated, even some kind of context help. If left unprocessed, interleaved sessions may adversely affect data quality, therefore this issue has to be dealt with. Here we have three choices: (i) neglect the problem, (ii) simply abandon such sessions, (iii) try to separate them. The first choice is bad for data quality since such sessions can affect web usage analysis results. If we abandon such sessions we also abandon useful knowledge about web site usage. Such sessions are usually generated by advanced users whose behavior could be potentially extremely valuable to us. Surprisingly, the interleaved session problem has been largely neglected in Web mining, although there is evidence that such users are actually among the better ones (e.g. better web shop buyers).

Data pre-processing is an important part of web usage analysis since it requires a lot of time and fundamentally affects the results of analyses. This problem motivated researchers to develop new methods for pre-processing. Colley et al. [2] propose a series of steps for data pre-processing for web usage mining. These include data cleaning, user identification, session identification and data formatting. Zhang et al. [11] improved statistical-based time oriented heuristics for the reconstruction of user sessions. They used statistical analysis and usage mining techniques to improve time-oriented heuristics. Ting et al. [9] developed the algorithm, which attempts to reconstruct missing server-side click-stream data based on referring site information and the Website’s link structure. Berendt et al. [1] used the web site structure to reconstruct incomplete sessions.

In [8] a basic process for separating interleaved sessions has been presented. Separation is based on first-order Markov model and sequential allocation of pages. Presented process takes into account the fact that transition between two pages $s_i \rightarrow s_{i+1}$ is more likely to belong to one of the consisting sessions. Each page $s_i$ in an interleaved session is sequentially allocated to one of the started separated sessions according to the probability of transition from the last page of separated session to the page $s_i$. The deficiency of the presented method is that it uses greedy approach which does not guarantee separated sessions with the highest probability of transition between consisting pages. The method was also limited to separation of interleaved session into up to two separated sessions only. They also state that, for large web sites with several thousands of web pages (possible states), the use of higher-order Markov models is discouraged because of increasing time and space complexity ($K$-th ordered polynomial).

We present an original approach for session separation based on data representation with first-order Markov models, upon which the separation process is build. To the very best of our knowledge, the Markov approach has not been used before for this particular purpose. The only remotely similar work concerning interleaved sessions is by Viermetz et al. [10] who work on building a user’s click-tree. Such a clicktree contains all possible paths a user could have taken through a website map. While their approach is dedicated to a better understanding of actual user behavior and not a separation process itself, our approach is focused on the separation process.
2 Methods

2.1 Evaluation of separating process

Let $S_1$ and $S_2$ be two HTTP sessions (sequences of Web pages). Let $S$ be an interleaved session produced by mixing $S_1$ and $S_2$. After separation we get from $S$ two separated sessions: $S'_1$ and $S'_2$. In order to evaluate the quality of separating process, we need to measure similarity between original sessions ($S_1$ and $S_2$) and sessions, produced with separation ($S'_1$ in $S'_2$):

$$\text{sim}(S_1; S_2, S'_1; S'_2) = \frac{\text{sim}(S_1, S'_1) + \text{sim}(S_2, S'_2)}{2}$$

(1)

Each session is represented as a sequence of pages. Evaluating quality of separated sessions can be viewed as evaluating their similarity. The problem lies in the proper definition of the similarity ($\text{sim}(S, S')$) measure. Determining the similarity between sequences is one of the basic tasks in machine translation as well as in computational biology [6]. Basically, two sequences are more similar if they have more symbols in common and the symbols' order is similar. There are many methods of measuring similarity between two sequences. We use several more or less strict methods based on: perfect match, Levenshtein distance, longest common subsequence (LCS) and weighted LCS [7].

2.2 Separating interleaved sessions with best-first search

In clickstream analysis, discrete Markov models are frequently used as analytical tools. In our approach discrete Markov models are only used as an appropriate clickstream data representation. The problem of separating interleaved sessions can be transformed into the problem of searching discrete Markov model space of partially separated interleaved sessions.

Generally speaking, a state space is a graph whose nodes correspond to problem situations, and finding a solution to the problem corresponds to finding a path in the graph. In the separating process we split the sequence of pages, (representing user sessions), into one or more shorter subsequences of pages. The process of assigning pages runs sequentially from the first to the last page in a sequence. Sequential order of pages in the separated sessions has to be the same as the order of pages in the interleaved session.

We begin the separating process with the original interleaved session of length $n$ and an list of $n$ empty separated sessions (the maximum possible number of sessions, where each state is in its own session). We finish with an empty interleaved session and a list of $k \leq n$ nonempty separated sessions. In each step we transfer the subsequent state (Web page) from the interleaved session to one of the growing separated sessions. Actually, we have to search for the optimal sequence of assignments of pages to the separated sessions.

The problem of separating sessions can be formulated as a state-space approach. Each state $Z$ represents interleaved session $S_P = [s_1, s_2, \ldots, s_n]$ of length $n$ and $r$ started separated sessions $S_R$. From one state to another we move with
legal actions (transitions with positive weight). Figure 1 shows an example of
two sample states and action, that transforms one state to another.

\[
\text{cost}(z_1\rightarrow z_2) = z_1 \cdot z_2
\]

\[
p(S_1\rightarrow S_8)
\]

Fig. 1. An example of a transition between two states in our search space.

Each action represents an assignment of the next page \( s_i \) from interleaved
session at the end of one of growing separated sessions \( S_R \). One of the possibilities
is also starting a new separated session with the number \( R + 1 \). We reach the
goal state, when the last page \( s_n \) from the interleaved session is assigned to one
of the started separated sessions \( S_R \). The leaves of the state space tree represent
end states. All leaves correspond to all partitions of the set of pages of the
interleaved session. We move from state to state at a certain cost. Every move
has attached a cost \( c(z_1, z_2) \), where \( c \) indicates the probability of transition from
the last state of one of the separated session \( S_R \) to the newly added page (state)
\( s_i \). If \( s_i \) indicates the beginning of a new session with the number \( r + 1 \), then
the cost of action equals \( c(z_1, z_2) = P_{Z,s_i}(s_i) \). The probability of each separated
session \( S_R \) is:

\[
P(S_R) = P_{Z,s_1}(sr_1) \prod_{i=1}^{a-1} P(sr_i \rightarrow sr_{i+1})
\] (2)

where \( P_{Z,s_i} \) denotes the probability that \( sr_1 \) is the starting page of a sequence and
\( a \) denotes the length the session. Let the function \( c(Z) \) be an estimator for each
node \( Z \) of the state space. We get the cost of node \( Z \) by multiplying the costs of
preceding actions along the path from the start node to the current node \( Z \). The
function \( c(Z) \) thus equals the product of probabilities of unfinished sessions:

\[
f(Z) = P(S_{R(1)}, S_{R(1)}, \ldots, S_{R(r)}) = \prod_{i=1}^{r} P(S_{R(i)})
\] (3)

The goal of the separating process is, according to interleaved session \( S_P \), to
find a goal node \( Z_G \) with the highest value of the estimator function \( c(Z_G) \). We
have to find the most probable (the cheapest in this sense) path from the start
node to one of possible goal node, which represents the sequence of page allocations
to separated sessions \( S_R \). The cost of the solution is the sum of edges along
the solution path. Unfortunately, due to the space and time complexity reasons,
it is impossible to use established graph algorithm (e.g. Dijkstra’s algorithm).
The size of the state space graph is too large. An algorithm that cuts down on
the size of the subgraph that must be explored, should be used instead.
Problem states connected with actions form directed graph that represent state space. Every state is connected with other states according to valid actions of partial interleaving.

Let us assume that from the state $Z$ we can transit to a set $\{Z_{next}\}$ of adjacent states by appending the page $s_i$ from the interleaved session to the end of one of the separated sessions $S_R$. Suppose state $Z$ has $n_r$ separated sessions. Since the first unallocated page $s_i$ of state $Z$ can be appended at the end of an existing session or start a new session, the state $Z$ leads to $n_r + 1$ new states ($Z_{next}$). It means that every state $Z$ has $n_r + 1$ successors. Every state in the set of successors $\{Z_{next}\}$ of state $Z$ is connected to $n_r + 1$ states as well. It is seen that the number of nodes in the state space increases rapidly with the height of the state space search tree.

If we have an interleaved session $S_P$ with $n$ pages, the state space for the problem of interleaving represents the tree $G$ with height $n$. With every allocated page $s_i$ from the interleaved session to one of the separated sessions, the height of the tree $G$ increases by 1. It can be proven that the number of nodes to the level $k$ correspond to Bell numbers $B_k$. Bell numbers can be acquired using recursion formula:

$$B_{n+1} = \sum_{k=0}^{n} \binom{n}{k} B_k$$

If we want to find the most probable session separation, we would principally have to search all possible states of the tree $G$. The number of all states in the tree $G$ is $\sum_{k=0}^{n} B_k$. Exhaustively searching all the states in graph $G$ can be extremely time consuming due to the problem of combinatorial complexity (exponential growth in $n$). Heuristic search aspires to solve this problem efficiently. The idea of heuristic search is to estimate which nodes are most promising in the candidate set and then continue searching in the way of the most promising node.

Traditional search method that is widely used is $A^*$ [3]. It falls into the class of priority-based search algorithms where only a small part of state space is searched in order to find goal node. Despite reduction of state space the order of complexity of $A^*$ is still exponential in a depth of search for time and space. Several variations of $A^*$ algorithm have been developed to save space, at the expense of time. We have chosen RBFS [5] because it uses linear space complexity and since we managed to find an efficient admissible heuristic function. We focus on the description of the heuristic function and outline the proof of its admissibility.

Let the heuristic estimator be a function $f(Z)$ that estimates the ‘difficulty’ of the state $Z$. The function $f$ estimates the cost of the best solution from the start to the goal node through node $Z$. The most promising candidate for the next node is the one that minimizes the function $f$. If the path goes through the node $Z$, the function $f$ can be written as the sum of two terms:

$$f(Z) = g(Z) + h(Z)$$

where $g(n)$ is an estimate of the cost of an optimal path from the start node to the node $Z$ and $h(Z)$ is the estimate of the best path from the node $Z$ to the
goal node. The calculation of the function $g(Z)$ is straightforward. The value of $g$ represents the cost of the current most optimal known path, that we traveled from the starting node to the node $Z$. Since for the session separating problem each action between two states represents probability of transition between two states, the value of the function $g(Z)$ equals to the product of probabilities of partially separated sessions.

Suppose we have in the node $Z$ state with $R$ started separated sessions. In order to use established best-first search approaches, we need to make the heuristic function both additive and positive. This can be achieved by applying the negative logarithm to the original (multiplicative) estimator function (Eq. 3). Negative logarithm of the function $g(Z)$ equals:

$$
\log g(Z) = -\log \sum_{r \in R} (P_{Z_S}(s_{r1})) + \sum_{r \in R} \sum_{i=1}^{\text{len}(S_r)} (-\log P(s_{ri} \rightarrow s_{ri+1})).
$$

The second term $h(Z)$ is more problematic because the nodes from $Z$ to the goal node have not been expanded yet by the search until this point. We do not know in advance how the other pages in the rest of the interleaved session (pages from $s_i$ to $s_n$) are going to be allocated to separated sessions. Therefore we do not know, what is the true cost of the path from the state $Z$ to the (as yet unknown) goal node. Since we do not know the page allocations, we obviously cannot calculate the probability of transitions between them. The function $h(Z)$ is therefore an estimate of the cost of the path from node $Z$ to the goal node. Therefore, $h(Z)$ is a heuristic guess, based on our knowledge of session separation problem. We constructed a non-trivial admissible heuristic function $h$ for the problem of separating interleaved sessions. Admissible heuristic function guarantees to find optimal solutions, that means the cheapest (most probable) path from start to goal node if the path exists. Best-first search which uses admissible function, finds the first goal node that is also the optimal one. Admissible functions are optimistic. That means that they underestimate the length of the path to the goal node. We considered this condition while constructing our heuristic function.

2.3 Heuristic function

Admissibility is a desired property of heuristic evaluation functions. Regarding our problem of separating interleaved sessions we have to find the goal node, which has the biggest product of transitions between pages in all separated sessions. Admissible heuristic function $h(Z)$ has to optimistically estimate the separation of the remaining part of the interleaved session. It has to maximize the product of the transition probabilities (and therefore minimize the sum of negative logarithms) between pages, that have not been assigned to the separated sessions yet. In the following part we will present heuristic for the problem of separating interleaved sessions. We will start with the trivial heuristic and continue with improved one that we used with the separation process. Developed heuristic function is admissible in case of starting pages of separated sessions can be determined with certainty. If separated session starting pages could not be determined reliably the separating process results in too many separated sessions.
A trivial heuristic admissible function for the problem of separating interleaved sessions is

\[ h_0(Z) = 1 \times 1 \ldots \times 1 = 1 \]

or (in negative logarithm space) \(-\log h_0(Z) = 0\). The trivial heuristic assumes that the probability of transition between remaining pages in the interleaved session is certain (equals 1). It complies to the condition for admissibility, since actual product of sequence probabilities for optimally separated sessions \( h^*(Z) \) is always smaller or equal to the value of \( h_0(Z) \). If we translate the product to sum of logarithms, we see, that it complies to well known condition for admissible heuristics \( h_0(Z) \leq h^*(Z) \).

We gradually improve the trivial heuristics with better one by utilizing knowledge about the search problem. The first improvement of heuristic function is to utilize the fact that the highest possible probability of transition into a particular page \( s_i \), \( \max\{P(\rightarrow s_i)\} \) is

\[
\max\{P(\rightarrow s_i)\} = \max_{j=1}^{N_s}\{P(s_j \rightarrow s_i)\},
\]

(5)

where \( N_s \) denotes the number of pages in the system. The second improvement is that in the calculation of the maximum probability of transition to page \( s_i \) \( (\max\{P(\rightarrow s_i)\}) \) we can take into account only those pages that reside in the interleaved session. The transition to the page \( s_i \) is possible only from those pages. We can cut down the number of pages that we take into account when calculating \( \max\{P(\rightarrow s_i)\} \) by considering the actual state of the separating process. Let us observe the node \( z3 \) in the Figure 2, where an interleaved session is partly separated in two started separated sessions. In the further separation process the page \( S5 \) will be either assigned to the existing session consisting of pages \( [S3, S0, S1] \), or it will follow one of the pages that have not yet been assigned and reside before page \( S5 \). When we calculate \( \max\{P(\rightarrow S5)\} \) we can take into account only pages \( S3, S0, S1 \) and \( S4 \).

Further we can taking into account the structure of already separated part of the interleaved session. The first page in figure 2 in the interleaved part of the session \( S4 \) can be only attached at the end of one of the current separated session. When we calculate \( \max\{P(\rightarrow s_i)\} \) we need not consider all the pages in the separated part but only the last page in each separated session. In case of node \( z3 \) and assignment of page \( S4 \) the expression is:

\[
\max\{P(\rightarrow S4)\} = \max\{P(S0 \rightarrow S4), P(S1 \rightarrow S4)\}
\]

The procedure for calculating the expression \( \max\{P(\rightarrow s_i)\} \) is depicted in figure 3. The condition for admissibility of heuristics \( h_1 \) is met according to the considerations outlined before, actual product of sequence probabilities for optimally separated sessions \( h^*(Z) \) is always smaller or equal to the value of \( h_1(Z) \). The difference between heuristic function \( h_1 \) and \( h_0 \) is better guidance in the state space for the new heuristic function. This heuristic function turned out to be efficient in guiding the search for the problem of session separation so we used it for separating real-world interleaved sessions.
3 Materials

First we created a test environment, similar to the real ones in terms of user session length and complexity. We created an artificial web site map, according to which artificial user sessions that were used for Markov chain training were generated. Another set of artificial sessions was generated in the same manner and used for creating interleaved sessions. After training the Markov chain (calculation of unconditional an transition probabilities), we applied the best-first search for separating interleaved sessions and verified the results. About 48% of interleaved sessions were separated 100% correctly (perfect match) and WLCS based average similarity was over 0.75, which encouraged us to proceed to real data.

We applied the interleaved session separating process on two real clickstream sources. The first clickstream originates from log files of university student records information system. It has been used by 16 member institutions. It has approximately 300 different pages. Each state in the Markov chain corresponds to an individual page. Typical user paths are well defined. Users have to be logged on in order to use the system, so determining the starting state is often (though not always) easy. Sometimes they are logged on with different user roles at the same time, and this creates interleaved sessions. Since users have to be logged on we can always determine the session entry point. The Web server log files use the basic CLF format. Clickstream data was taken for 4 months of use, which resulted in 150,000 user sessions.

The second clickstream source is taken from a large web shop, which is considerably different from the student records information system. Users do not have to sign in (except for buying items), it has many more users and many more
We had to cut down number of states of Markov model in order to efficiently use it. Every state of our Markov model represents a group of pages, not an individual page. We transformed the web shop pages to 900 states. Session entry point can be almost any page, which makes separating interleaved sessions harder. The site map has plenty of links between pages. In fact only few pages are not linked with all others. It generates about 10,000 user sessions a day.

For both clickstreams we took the same steps as with artificially generated data. After separating interleaved sessions we evaluated results with evaluation methods that we presented earlier.

4 Results

We tested the process of separating interleaved sessions on artificially generated interleaved sessions made from real-world clean sessions. Interleaved sessions were created by interleaving clean sessions having in mind the properties and structure of typical real interleaved sessions. The advantage of artificially generated sessions in comparison with actual interleaved sessions is the knowledge of elementary clean sessions that form interleaved sessions. This makes it possible to evaluate the quality of separation process.

For the generation of interleaved sessions only the clean sessions with length between 4 and 100 user page requests were used (shorter and longer sessions are mostly web robots and artefacts). Generated interleaved sessions can consist from various number of clean sessions. According to the results of analysis of student records IS, the majority of interleaved sessions consist out of two or three clean sessions. Sessions with more interleaving are very rare. We created a test set consisting of interleaved sessions that contain one (clean session), two or three interleaving clean sessions. The number of interleaved sessions in each of three groups is equal, so we have one third of sessions for each group of interleaved sessions. In reality the number of interleaved sessions decline exponentially with the number of contained clean sessions. We have 98.8% of clean sessions, 1% interleaved sessions with two and 0.1% with three interleaving clean sessions.

Table 1 presents results of separation process using RBFS. The first row of the table contains results for student records IS and the second row results for web shop. Values in the table represent $F$-measure based average similarity between separated and actual sessions. Values can range from 0 (totally different) to 1 (completely equal). Each column of the table corresponds to one method for results evaluation. The second row of the table contains evaluation results according to the perfect match. Sessions can be separated 100% correctly (session sequence similarity = 1) or separated incorrectly (value 0). According to this evaluation method one badly positioned page in one of the separated sessions means already incorrect separation. Average similarity for this method means the proportion of all 100% correctly separated sessions with respect to all processed interleaved sessions. Hence, a result of 0.5 in the first row and the second column means that 50% of the interleaved sessions have been separated 100% correctly. Other evaluation methods also result in values between 0 and 1.
Table 1. The results of separating for both clickstream data sources using RBFS.

<table>
<thead>
<tr>
<th>source</th>
<th>complete equality</th>
<th>edit-distance</th>
<th>LCS</th>
<th>WLCS</th>
<th>skip-bigram</th>
</tr>
</thead>
<tbody>
<tr>
<td>e-Student</td>
<td>0.500</td>
<td>0.743</td>
<td>0.781</td>
<td>0.755</td>
<td>0.714</td>
</tr>
<tr>
<td>Web shop</td>
<td>0.405</td>
<td>0.565</td>
<td>0.596</td>
<td>0.567</td>
<td>0.543</td>
</tr>
</tbody>
</table>

Results in the table 1 show that we achieved better results for student record IS than for the web shop. We have separated entirely correct 50% of user sessions for student records IS and around 40% of interleaved sessions for the web shop. If we have a look at other methods of evaluation, the ration between results for both systems is similar. User sessions of student records IS have more defined structure, we can also more easily detect starting pages of consisting clean sessions. One of the reasons for worse results for web shop is that we used groups of pages instead of pages for Markov model. Grouping pages together affects the results. Since the web shop site map is larger, there may be numerous user paths, that also affects the results. User can enter the web shop at almost any page, so it is harder to detect where the second session in interleaved session starts. All this reflects in better results for student records IS.

Figures 4 and 5 shows graphs for two different methods of results evaluation. Figure 4 reports results for student IS clickstream while Figure 5 reports results for web shop. Each graph corresponds to one evaluation method. The X axis presents intervals for perfect match or F-measure based sequence similarity, while the Y axis shows the number of separated sessions that fall in the similarity interval. Above each column the percentage of sessions in that interval is presented. A dotted line in each graph represents the median session similarity for corresponding similarity measure. Better results of separations are the one that have as many separations as possible closer to 1. Ideal results would be presented with a graph that has just one column in the interval of session similarity between 0.9 and 1.

The first graph in both figures depicts the first column in the table 1, that is, the number of perfectly matched separated sessions. For student records IS 50% of interleaved sessions have been separated 100% correctly (session sequence similarity = 1) and for web shop 40%. The second graphs in both figures depict results according to the WLCS evaluation method. Comparison of WLCS graphs for both clickstreams shows that the quality of interleaved sessions separation is much better for student records IS than web shop. Let us focus only to the interval of similarity between 0.1 and 0.9 for WLCS evaluation method. The distribution of separated sessions with the similarity interval between 0.1 and 0.9 is quite similar to both clickstream data sources. The biggest difference is for the separated sessions with the similarity between 0 and 0.1. This are the separated sessions that have very little in common with the actual clean sessions that construe interleaved sessions, so we would like to cut down their number as much as possible. For the student records IS there are three times less badly separated sessions (10%) than for web shop (30,3%).

72
5 Conclusion

We propose a new method for improving the quality of clickstream data in pre-processing phase that is based on the first-order Markov model and best-first heuristic search. We present the motivation that led us to implementation, and apply the method in practice on two real data clickstreams. The proposed method proved to be quite successful. It gives promising results and is independent of the number of clean sessions that create an interleaved session. Since the method automatically determines the most probable number of interleaved sessions, the highest possible amount of interleaving is not a method parameter, and this increases the value of achieved results. The developed method can be used on various clickstream data sources, however in general better results are expected if user sessions are well defined.
The presented method works on the basis of the most probable sequence of pages. It has higher space and time complexity than the simple greedy method presented in [8] but can be used more generally. Presented method uses linear space complexity according to the interleaved session length. In general it has exponential time complexity but it also depends on the quality of heuristic function. Our method alleviates the problem of interleaved HTTP sessions and improves the ETL process quality for clickstream data. The net result are better data in a data webhouse.

In the further work we will focus on improvement of session separation accuracy. Proper identification of start pages is the key to quality separation process so we will focus on better identification of start pages of separated sessions. We will investigate ways to lower average time complexity. One possibility is upgrading the separation process with Real-Time-A* (RTA*) [4].

References

Abstract. According to Koestler, the notion of a bisocation denotes a connection between pieces of information from habitually separated domains or categories. In a former paper, we presented a methodology to find such bisociations using a network representation of knowledge, which is called a BisoNet, because it promises to contain bisociations. To achieve this, a custom evaluation measure for the links in the network, called the "Bison measure", has been introduced, and experiments have shown that it has good properties for bisociation discovery. In this paper, we will present an other methodology to find bisociations, this time looking into databases for promising item sets, using a frequent item set mining technique, coupled with an extension of the Bison measure used formerly in the binary bisociations search. In a first step, we will show how the Bison measure has been adapted to fulfill the requirements of the frequent item set mining technique, highlighting how we can reach, starting from the minimum T-norm, a measure that keeps most of the Bison measure’s properties and satisfies the requirements of the frequent item set mining algorithm, such as that it should be downward closed. In a second step, we will perform some experiments on data sets containing informations from different domains to evaluate the ability of our method to find bisociative item sets, showing that interesting data sets are found. We will also discuss how can the quality of the results be evaluated in an automatic way, that is to say without the need of an expert to have a look at each set to decide if it is an interesting one or not, taking into account how many different domains are included while building item sets. Future work includes validating the relevancy of the item sets found according to real human knowledge and working on domains’ relations to each other, extending our methods to non-textual data and improving the similarity measure.

1 Introduction

The concept of association is at the heart of many of today’s powerful ICT technologies such as information retrieval and data mining. These technologies typically employ “association by similarity or co-occurrence” in order to discover new information that is relevant to the evidence already known to a user.
However, domains that are characterized by the need to develop innovative solutions require a form of creative information discovery from increasingly complex, heterogeneous and geographically distributed information sources. These domains, including design and engineering (drugs, materials, processes, devices), areas involving art (fashion and entertainment), and scientific discovery disciplines, require a different ICT paradigm that can help users to uncover, select, re-shuffle, and combine diverse contents to synthesize new features and properties leading to creative solutions. People working in these areas employ creative thinking to connect seemingly unrelated information, for example, by using metaphors or analogical reasoning. These modes of thinking allow the mixing of conceptual categories and contexts, which are normally separated. The functional basis for these modes is a mechanism called *bisociation* (see [1]).

According to Arthur Koestler, who coined this term, *bisociation* means to join unrelated, and often even conflicting, information in a new way. It means being “double minded” or able to think on more than one plane of thought simultaneously. Similarly, Frank Barron [2] says that the ability to tolerate chaos or seemingly opposite information is characteristic of creative individuals.

Several famous scientific discoveries are good examples of bisociations, for instance Isaac Newton’s theory of gravitation and James C. Maxwell’s theory of electromagnetic waves. Before Newton, a clear distinction was made between *sub-lunar* (below the moon) and *super-lunar physics* (above the moon), since it was commonly believed that these two spheres were governed by entirely different sets of physical laws. Newton’s insight that the trajectories of planets and comets can be interpreted in the same way as the course of a falling body joined these habitually separated domains. Maxwell, by realizing that light is an electromagnetic wave, joined the domains of optics and electromagnetism, which, at his time, were also treated as unrelated areas of physical phenomena.

Although the concept of bisociation is frequently discussed in cognitive science, psychology and related areas (see, for example, [1–3]), there does not seem to exist a serious attempt at trying to formalize and computerize this concept. In terms of ICT implementations, much more widely researched areas include association rule learning (for instance, [4]), analogical reasoning (for example, [5, 6]), metaphorical reasoning (for example, [7]), and related areas such as case-based reasoning (for instance, [8]) and hybrid approaches (for example, [9]).

In order to fill this gap in current research efforts, the BISON project\(^1\) was created. This project focuses on a knowledge representation approach with the help of networks of named entities, in which bisociations may be revealed by link discovery and graph mining methods, but also by computer-aided interactive navigation. In this paper we report first results obtained in this project.

The rest of this paper is structured as follows: in Section 2 we provide a definition of the core notion of a *bisociation*, which guides our considerations. Section 3 presents how we apply frequent item sets mining to the discovery of bisociative item sets, according to the definition of bisociation we presented. In

\(^1\) See [http://www.bisonet.eu/](http://www.bisonet.eu/) for more information on this EU FP7 funded project.
particular, we present a new measure for the quality of the item sets found, based on the Bison measure used in former works on Bisonets presented in [10].

Afterwards, in Section 4 we report results on a benchmark data set. By showing that with our system we can produce item sets reaching good quality values from this data source, we provide evidence that the computer-aided search for bisociations is a highly promising technology.

Finally, in Section 5 we draw conclusions from our discussion.

2 Bisociation

Since the core notion of our efforts is *bisociation*, we start by trying to provide a sufficiently clear definition, which can guide us in our attempts to create a system able to support a user in finding bisociations. A first definition within the BISON project\(^2\) characterizes *bisociation* as follows:

A *bisociation* is a link \(L\) that connects two domains \(D_1\) and \(D_2\) that are unconnected given a specific context or view \(V\) by which the domains are defined. The link \(L\) is defined by a connection between two concepts \(c_1\) and \(c_2\) of the respective domains.

Although the focus on a connection between two habitually (that is, in the context a user is working in) separated domains is understandable, this definition seems somewhat too narrow. Linking two concepts from the same domain, which are unconnected within the domain, but become connected by employing indirect relations that pass through another domain, may just as well be seen as bisociations. The principle should rather be that the connection is not fully contained in one domain (which would merely be an association), but needs access to a separate domain. Taking this into account, we generalize the definition:

A *bisociation* is a link \(L\) between two concepts \(c_1\) and \(c_2\), which are unconnected given a specific context or view \(V\). The concepts \(c_1\) and \(c_2\) may be unconnected, because they reside in different domains \(D_1\) and \(D_2\) (which are seen as unrelated in the view \(V\)), or because they reside in the same domain \(D_1\), in which they are unconnected, and their relation is revealed only through a *bridging concept* \(c_3\) residing in some other domain \(D_2\) (which is not considered in the view \(V\)).

In both of these characterizations we define domains formally as sets of concepts. Note that a *bridging concept* \(c_3\) is usually also required if the two concepts \(c_1\) and \(c_2\) reside in different domains, since direct connections between them, even if they cross the border between two domains, can be expected to be known and thus will not be interesting or relevant for a user.

Starting from the above characterization of *bisociation*, a network representation, called a *BisoNet*, of the available knowledge suggests itself: each concept

\(^2\) See http://www.inf.uni-konstanz.de/bisonwiki/index.php5, which, however, is not publicly accessible at this time.
(or, more generally, any named entity) gives rise to a node. Concepts that are associated (according to the classical paradigm of similarity or co-occurrence) are connected by an edge. Bisociations are then indirect connections (technically paths) between concepts, which cross the border between two domains.

Note that this fits both forms of bisociations outlined above. If the concepts \( c_1 \) and \( c_2 \) reside in different domains, the boundary between these two domains necessarily has to be crossed. If they reside in the same domain, one first has to leave this domain and then come back in order to find a bisociation.

3 Frequent Item Sets Mining for Bisociations

To build the item set, we use a frequent item set mining algorithm, called “xfim” for extended frequent item set mining, that build frequent item sets using their co-occurrences in documents. We will therefore introduce frequent item set mining in a first time.

Once these item sets are built, they need to be evaluated in order to determine if they are promising to contain bisociations. For this purpose, we will present an adaptation to frequent item sets mining of the Bison measure presented in [10].

3.1 Frequent Item Sets Mining principles

Frequent item set mining is a data analysis method, which was originally developed for market basket analysis and which aims at finding regularities in the shopping behavior of the customers of supermarkets, mail-order companies and online shops. In particular, it tries to identify sets of products that are frequently bought together. Once identified, such sets of associated products may be used to optimize the organization of the offered products on the shelves of a supermarket or the pages of a mail-order catalog or web shop, or may give hints which products may conveniently be bundled.

Formally, the task of frequent item set mining can be described as follows: we are given a set \( B \) of items, called the item base, and a database \( T \) of transactions. Each item represents a product, and the item base represents the set of all products offered by a store. The term item set refers to any subset of the item base \( B \). Each transaction is an item set and represents a set of products that has been bought by an actual customer. Since two or even more customers may have bought the exact same set of products, the total of all transactions must be represented as a vector, a bag or a multiset, since in a simple set each transaction could occur at most once.\(^3\) Note that the item base \( B \) is usually not given explicitly, but only implicitly as the union of all transactions.

The support \( s_T(I) \) of an item set \( I \subseteq B \) is the number of transactions in the database \( T \), it is contained in. Given a user-specified minimum support \( s_{\text{min}} \in \mathbb{N} \), an item set \( I \) is called frequent in \( T \) iff \( s_T(I) \geq s_{\text{min}} \). The goal of

\(^3\) Alternatively, each transaction may be enhanced by a unique transaction identifier, and these enhanced transactions may then be combined in a simple set.
frequent item set mining is to identify all item sets $I \subseteq B$ that are frequent in a given transaction database $T$. Note that the task of frequent item set mining may also be defined with a relative minimum support, which is the fraction of transactions in $T$ that must contain an item set $I$ in order to make $I$ frequent. However, this alternative definition is obviously equivalent.

A standard approach to find all frequent item sets w.r.t. a given database $T$ and support threshold $s_{\text{min}}$, which is adopted by basically all frequent item set mining algorithms (except those of the Apriori family), is a depth-first search in the subset lattice of the item base $B$. Viewed properly, this approach can be interpreted as a simple divide-and-conquer scheme. For some chosen item $i$, the problem to find all frequent item sets is split into two subproblems: (1) find all frequent item sets containing the item $i$ and (2) find all frequent item sets not containing the item $i$. Each subproblem is then further divided based on another item $j$: find all frequent item sets containing (1.1) both items $i$ and $j$, (1.2) item $i$, but not $j$, (2.1) item $j$, but not $i$, (2.2) neither item $i$ nor $j$, and so on.

All subproblems that occur in this divide-and-conquer recursion can be defined by a conditional transaction database and a prefix. The prefix is a set of items that has to be added to all frequent item sets that are discovered in the conditional database. Formally, all subproblems are tuples $S = (C, P)$, where $C$ is a conditional database and $P \subseteq B$ is a prefix. The initial problem, with which the recursion is started, is $S = (T, \emptyset)$, where $T$ is the transaction database to mine and the prefix is empty. A subproblem $S_0 = (C_0, P_0)$ is processed as follows: Choose an item $i \in B_0$, where $B_0$ is the set of items occurring in $C_0$. This choice is arbitrary, but usually follows some predefined order of the items. If $s_{C_0}(i) \geq s_{\text{min}}$, then report the item set $P_0 \cup \{i\}$ as frequent with the support $s_{C_0}(i)$, and form the subproblem $S_1 = (C_1, P_1)$ with $P_1 = P_0 \cup \{i\}$. The conditional database $C_1$ comprises all transactions in $C_0$ that contain the item $i$, but with the item $i$ removed. This also implies that transactions that contain no other item than $i$ are entirely removed: no empty transactions are ever kept. If $C_1$ is not empty, process $S_1$ recursively. In any case (that is, regardless of whether $s_{C_0}(i) \geq s_{\text{min}}$ or not), form the subproblem $S_2 = (C_2, P_2)$, where $P_2 = P_0$ and the conditional database $C_2$ comprises all transactions in $C_0$ (including those that do not contain the item $i$), but again with the item $i$ removed. If $C_2$ is not empty, process $S_2$ recursively.

Eclat, FP-growth, RElim and several other frequent item set mining algorithms rely on this basic scheme, but differ in how they represent the conditional databases. The main approaches are horizontal and vertical representations. In a horizontal representation, the database is stored as a list (or array) of transactions, each of which is a list (or array) of the items contained in it. In a vertical representation, a database is represented by first referring with a list (or array) to the different items. For each item a list (or array) of identifiers is stored, which indicate the transactions that contain the item.

However, this distinction is not pure, since there are many algorithms that use a combination of the two forms of representing a database. For example, while Eclat uses a purely vertical representation, FP-growth combines in its FP-
tree structure a vertical representation (links between branches) and a (compressed) horizontal representation (prefix tree of transactions). RElim uses basically a horizontal representation, but groups transactions w.r.t. their leading item, which is, at least partially, a vertical representation.

More detailed description of the various algorithms talked about here can be found in [11].

3.2 The Bison measure

In [10], we present in details how we achieved to build the Bison measure for two sets of TF values, willing to take into account the term frequency values itself (and not only their difference) by using the product of the two term frequency values as a coefficient to the (absolute) difference between the term frequency values. This takes care of the fact that the two term frequency values have to be positive, and that the similarity value should be the greater, the larger the term frequency values are (and, of course, the smaller their absolute difference is).

However, in our case, we also want to take into account that it is better to have two similar term frequency values of 0.35 (which means that the two keywords both appear rather infrequently in the document) than to have term frequency values of 0.3 and 0.7 (which means the first keywords appears rarely, while the other quite frequently).

In order to adapt the product to this consideration, we use the expression in Equation 1, in which \( k \) can be adjusted according to the importance one is willing to give to low term frequency values.

\[
B(A, B) = (tf^A_i \cdot tf^B_i)^k \cdot (1 - |tf^A_i - tf^B_i|), \quad tf^A_i, tf^B_i \in [0, 1]
\]

(1)

Still another thing that we have to take into account in our case is that the same difference between \( tf^A_i \) and \( tf^B_i \) can have a different impact depending on whether \( tf^A_i \) and \( tf^B_i \) are large or small. To tackle this issue, we combine Equation 1 with the use of the arctan function, and thus obtain the similarity measure shown in Equation 2, which we call the Bison measure. This form has the advantage that it takes into account that two term frequency values for the same index have to be positive, that the similarity should be the greater, the larger the term frequency values are, and that the same difference between \( tf^A_i \) and \( tf^B_i \) should have a different impact according to the values of \( tf^A_i \) and \( tf^B_i \).

\[
B(A, B) = (tf^A_i \cdot tf^B_i)^k \cdot \left(1 - \frac{|\arctan(tf^A_i) - \arctan(tf^B_i)|}{\arctan(1)}\right), \quad tf^A_i, tf^B_i \in [0, 1]
\]

(2)

3.3 Adapting the Bison measure to frequent item set mining

As the Bison measure has shown to be effective in finding interesting links in Bisonets, we here want to use the same properties to select our frequent item sets, although we cannot use the Bison measure as it is due to requirements of
the mining algorithm used, for example the measure has to be monotonous and downward closed, which the Bison measure is not.

A first idea was to have a look to the T-norm measures which are quite standard in sets comparisons. Plotting the Bison measure and some T-Norm measures showed us that the minimum t-norm measure was quite close the the Bison one (see Figure 1).

Although this minimum t-norm looks quite similar to the Bison measure, we wanted to keep some interesting specificities of the Bison measure, namely that the co-occurrence of two terms that have rather low but similar term frequency values should be rated higher than the co-occurrence of a term with a low value and a term with a high value.

For this purpose, we started with the minimum t-norm and introduced a second term and exponents on both terms, as shown in Equation 3 to be able to reflect this Bison measure’s behavior without loosing the properties needed by the algorithm to end with a custom measure that we call the xfim measure.

\[
x_{fim}^{a,b}(x, y) = \min(x, y)^a \cdot (1 - \max(x, y) + \min(x, y))^b, \quad a \geq 0, b \geq 0. \tag{3}
\]

The measure \(x_{fim}^{a,b}(x, y)\) is anti-monotone: both \(\min(x, y)\) and \(1 - \max(x, y)\) are obviously anti-monotone. In addition, the sum and the product of two anti-monotone measures is anti-monotone and a monotone transformation of an anti-monotone measure is anti-monotone. Therefore \(x_{fim}^{a,b}(x, y)\) must be anti-monotone, because it is constructed from \(\min(x, y)\) and \(1 - \max(x, y)\) by summing, multiplying and applying monotone transformations.

4 The Wikipedia-Schools benchmark

Having shown how BisoNets can be built from textual data sources, we present a benchmark application in this section. The idea is to provide a proof of principle, that this approach of finding frequent item sets using the xfim measure can help a user to discover bisociations.

Schools Wikipedia is a free, hand-checked, non-commercial selection from Wikipedia, targeted around the UK National Curriculum and useful for much of the English speaking world. It has about 5500 articles (as much as can be fitted on a DVD with good size images) and is about the size of a twenty volume encyclopedia (34.000 images and 20 million words). Wikipedia is the free encyclopedia anyone can edit, and develops accurate content but suffers vandalism. Wikipedia is not necessarily a child-safe environment and has "adult" content. This selection of topics have been carefully chosen, tidied up, and checked for vandalism and suitability (by SOS Children volunteers, whom we gratefully acknowledge).

Our data is composed of a file listing, for a given document, all the keywords that occur with their respecting TFIDF value, and another file listing, for a given document, all the categories this document belongs to.
Fig. 1. Plotting different T-norm measures and the Bison measure. The Minimum T-norm seems the closest to the Bison measure.
The first file can be used as it is as an input for the xfim program, each line representing a transaction. Frequent item sets are then being built and the xfim value computed for each item set, using Equation 3.

An interesting thing to see is then how many domains are included in each item set, and for this, we need some processing using the 2 data files we have, assigning to each term a list of domains, and then, processing the output of the xfim program to assign each item set a list of domains.

In order to really catch how a term is able to link domains between each other, we did not just use the number of domains that are included in an item set, but we compute, for each term, a measure, the “domains measure”, which is the total number of domains associated with a term, divided by the average number of domains associated with a document containing the term. Then, the quality measure for an item set is the average of this measure.

The experiment has been done setting the xfim program to produce item sets of 3 or more items, no minimum number of supports (documents) included in the item sets produced and a minimum normalized xfim value for the item sets of 0.015. The characteristics of the item sets produced are shown in Table 1.

<table>
<thead>
<tr>
<th>Number of item sets</th>
<th>4980</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max. number of documents in an item set</td>
<td>79</td>
</tr>
<tr>
<td>Min. number of documents in an item set</td>
<td>12</td>
</tr>
<tr>
<td>Max. domains measure value for an item set</td>
<td>12.58</td>
</tr>
<tr>
<td>Min. domains measure value for an item set</td>
<td>1.63</td>
</tr>
<tr>
<td>Correlation between number of documents and normalized xfim</td>
<td>0.004933</td>
</tr>
<tr>
<td>Correlation between number of documents and domains measure</td>
<td>0.002628</td>
</tr>
<tr>
<td>Correlation between normalized xfim and domains measure</td>
<td>0.09130</td>
</tr>
</tbody>
</table>

Table 1. Summary of the characteristics of the item sets produced.

In Table 1 and in Figure 2, we can see that the number of documents, the normalized xfim measure and the domains measure are very weakly correlated, thus we have three different independent measure to rate the item sets.

We can see in these results that the item sets produced, when considering for example the 50 best according to the normalized xfim measure, have an average domains measure of 7.12, which means they promise to be linking several domains, and therefore might be containing some interesting domain-linking terms (see Table 2 for the detailed item sets).

5 Conclusion and further work

In this article, we provided a definition of the notion of a bisociation, as understood by Koestler, which is the key notion of the BISON project. Building on this definition, we then defined the concept of a BisoNet, which is a network bringing together data sources from different domains, and therefore may help...
Fig. 2. Graphics showing the independence of the different measures used to evaluate the item sets.
<table>
<thead>
<tr>
<th>Items</th>
<th>Normalized xfm measure</th>
<th>domains measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>class genu bird</td>
<td>0.14463</td>
<td>8.40982</td>
</tr>
<tr>
<td>tree genu import</td>
<td>0.13667</td>
<td>7.50494</td>
</tr>
<tr>
<td>class genu import</td>
<td>0.13535</td>
<td>10.0288</td>
</tr>
<tr>
<td>class genu speci</td>
<td>0.13134</td>
<td>8.30169</td>
</tr>
<tr>
<td>tree genu bird</td>
<td>0.13126</td>
<td>5.88601</td>
</tr>
<tr>
<td>tree genu import speci</td>
<td>0.13054</td>
<td>7.75305</td>
</tr>
<tr>
<td>genu import speci</td>
<td>0.13036</td>
<td>8.67074</td>
</tr>
<tr>
<td>class genu import speci</td>
<td>0.12961</td>
<td>9.64591</td>
</tr>
<tr>
<td>tree genu speci</td>
<td>0.12863</td>
<td>5.77788</td>
</tr>
<tr>
<td>class genu bird speci</td>
<td>0.12796</td>
<td>8.43171</td>
</tr>
<tr>
<td>scientif import speci</td>
<td>0.12793</td>
<td>8.02691</td>
</tr>
<tr>
<td>class bird import</td>
<td>0.12776</td>
<td>11.6906</td>
</tr>
<tr>
<td>scientif genu bird</td>
<td>0.12745</td>
<td>4.85427</td>
</tr>
<tr>
<td>class bird import speci</td>
<td>0.12568</td>
<td>10.8923</td>
</tr>
<tr>
<td>scientif genu import</td>
<td>0.12550</td>
<td>6.47320</td>
</tr>
<tr>
<td>class import speci</td>
<td>0.12525</td>
<td>11.5825</td>
</tr>
<tr>
<td>tree import speci</td>
<td>0.12496</td>
<td>9.05865</td>
</tr>
<tr>
<td>scientif genu import speci</td>
<td>0.12480</td>
<td>6.97924</td>
</tr>
<tr>
<td>class bird speci</td>
<td>0.12398</td>
<td>9.96353</td>
</tr>
<tr>
<td>passerin import speci</td>
<td>0.12345</td>
<td>8.05865</td>
</tr>
<tr>
<td>insect genu import</td>
<td>0.12287</td>
<td>7.11682</td>
</tr>
<tr>
<td>scientif genu speci</td>
<td>0.12274</td>
<td>4.74613</td>
</tr>
<tr>
<td>insect genu bird</td>
<td>0.1225</td>
<td>5.49789</td>
</tr>
<tr>
<td>passeriform genu bird</td>
<td>0.12200</td>
<td>4.55268</td>
</tr>
<tr>
<td>passeriform genu import</td>
<td>0.12200</td>
<td>6.17161</td>
</tr>
<tr>
<td>passeriform genu speci</td>
<td>0.12200</td>
<td>4.44455</td>
</tr>
<tr>
<td>observ star planet</td>
<td>0.12149</td>
<td>3.89535</td>
</tr>
<tr>
<td>scientif bird import</td>
<td>0.12080</td>
<td>8.13504</td>
</tr>
<tr>
<td>scientif bird speci</td>
<td>0.12080</td>
<td>6.40798</td>
</tr>
<tr>
<td>woodland genu speci</td>
<td>0.12080</td>
<td>5.05566</td>
</tr>
<tr>
<td>insect import speci</td>
<td>0.12078</td>
<td>8.67052</td>
</tr>
<tr>
<td>underpart import speci</td>
<td>0.12042</td>
<td>7.72532</td>
</tr>
<tr>
<td>insect genu import speci</td>
<td>0.12026</td>
<td>7.46196</td>
</tr>
<tr>
<td>genu bird import</td>
<td>0.11896</td>
<td>8.77887</td>
</tr>
<tr>
<td>specimen bone dinosaur</td>
<td>0.11880</td>
<td>5.16528</td>
</tr>
<tr>
<td>bone dinosaur format</td>
<td>0.11823</td>
<td>6.46190</td>
</tr>
<tr>
<td>animalia genu bird</td>
<td>0.11813</td>
<td>4.86581</td>
</tr>
<tr>
<td>insect genu speci</td>
<td>0.11810</td>
<td>5.38975</td>
</tr>
<tr>
<td>conserv genu bird</td>
<td>0.11801</td>
<td>6.16379</td>
</tr>
<tr>
<td>scientif bird import speci</td>
<td>0.11749</td>
<td>8.22562</td>
</tr>
<tr>
<td>genu bird import speci</td>
<td>0.11742</td>
<td>8.70850</td>
</tr>
<tr>
<td>scientif phylum speci</td>
<td>0.11737</td>
<td>4.44357</td>
</tr>
<tr>
<td>run test match</td>
<td>0.11732</td>
<td>3.62745</td>
</tr>
<tr>
<td>passeriform genu import speci</td>
<td>0.11731</td>
<td>6.75305</td>
</tr>
<tr>
<td>song import speci</td>
<td>0.11716</td>
<td>7.72532</td>
</tr>
<tr>
<td>bone dinosaur genu</td>
<td>0.11694</td>
<td>4.81209</td>
</tr>
<tr>
<td>plumag import speci</td>
<td>0.11657</td>
<td>8.03566</td>
</tr>
<tr>
<td>tree bird import</td>
<td>0.11626</td>
<td>9.16678</td>
</tr>
<tr>
<td>run test team</td>
<td>0.11591</td>
<td>5.21429</td>
</tr>
<tr>
<td>tenni singl championship</td>
<td>0.11590</td>
<td>4.83065</td>
</tr>
</tbody>
</table>

Table 2. Details of the 50 best item sets produced according to the adapted Bison measure.
a user to discover bisociations. We presented a way to produce frequent item sets using a customized frequent item set mining, introducing an appropriate measure to rate the item sets.

We then tested our approach on a benchmark in order to produce item sets that would be bridging different domains in a bisociation way.

In summary, we venture to say that this work can be easily applied to any kind of textual data source in order to mine data looking for interesting, item sets. Further work also consists in performing other benchmarks, find methods to rate the relevancy of the item sets found, and applying other quality measures for the item sets that would lead to better results in a bisociation point of view.

The work presented here was supported by the European Commission under the 7th Framework Program FP7-ICT-2007-C FET-Open, contract no. BISON-211898.

References

Unfolding network communities by combining defensive and offensive label propagation

Lovro Šubelj* and Marko Bajec

Laboratory for Data Technologies, University of Ljubljana, Ljubljana, Slovenia

Abstract. Label propagation has proven to be a fast method for detecting communities in complex networks. Recent work has also improved the accuracy and stability of the basic algorithm, however, a general approach is still an open issue. We propose different label propagation algorithms that convey two unique strategies of community formation, namely, defensive preservation and offensive expansion of communities. Furthermore, the strategies are combined in an advanced label propagation algorithm that retains the advantages of both approaches; and are enhanced with hierarchical community extraction, prominent for the use on larger networks. The proposed algorithms were empirically evaluated on different benchmarks networks with planted partition and on over 30 real-world networks of various types and sizes. The results confirm the adequacy of the propositions and give promising grounds for future analysis of (large) complex networks. Nevertheless, the main contribution of this work is in showing that different types of networks (with different topological properties) favor different strategies of community formation.

Keywords: Network communities, label propagation, defensive preservation, offensive expansion.

1 Introduction

Complex networks commonly comprise of local structural modules or communities that are groups of nodes strongly connected within and only weakly connected with the rest of the network. These modules play crucial roles in many real-world systems [15, 37], moreover, they provide an important insight into structure and function of (large) complex networks [37, 45, 27].

Over the last decade the research community has shown a considerable interest in detecting communities in real-world networks. Thus, a number of approaches has been presented in the literature. In particular, approaches optimizing modularity\(^1\) \(Q\) [7, 6, 5], graph partitioning [14, 39, 38] and spectral [9, 33] algorithms, statistical methods [36], algorithms based on dynamic processes [40, 43, 38, 42], overlapping, hierarchical and multiresolution methods [37, 16, 42], and other [29, 30] (for a thorough review see [11]).

* Corresponding author: lovro.subelj@fri.uni-lj.si.
\(^1\) Significance of communities due to a selected null model [35].
Due to the size of large real-world networks recent research has focused on developing scalable algorithms that can be applied to networks with several millions of nodes and billions of edges. Raghavan et al. [40] proposed using simple label propagation, where labels are propagated among nodes until an equilibrium is reached. The main advantage of the label propagation is its near linear time complexity (in the number of edges of the network); however, due to the algorithm’s simplicity, the accuracy of revealed community structure is often not state-of-the-art.

The basic algorithm was further analyzed in [47] and refined into a modularity optimization algorithm in [5, 29]. Extension to directed networks was considered in [28]. Furthermore, Leung et al. [28] improved the basic label propagation by applying label hop attenuation and node preference (i.e. node propagation strength). We proceed their work in developing two unique strategies of community formation, namely, defensive preservation of communities, where preference is applied to the core of each community, and offensive expansion of communities, where preference is applied to the border of each community. Moreover, the two strategies are combined into an advanced label propagation algorithm (denoted K-Cores) that preserves the advantages of both approaches. For the use with larger networks, we also present two different manners of hierarchical community extraction.²

Proposed algorithms were rigorously analyzed on different benchmark networks with planted partition and on a large number of real-world networks of various types and sizes. The results justify the adequacy of the propositions and give promising grounds for future analysis of (large) complex networks. Furthermore, the analysis also shows that the appropriateness of the strategies of community formation strongly correlates with the type of the network (i.e. with its topological properties).

The rest of the article is structured as follows. Section 2 gives a formal presentation of label propagation and briefly surveys relevant subsequent refinements of the basic algorithm. Defensive and offensive strategies of community formation, and corresponding algorithms, are presented and discussed in section 3. Empirical evaluation with discussion is done in section 4 and conclusion in section 5.

2 Label propagation and advances

Let the network be represented by an undirected (multi-)graph \( G(N, E) \), where \( N \) is the set of nodes and \( E \) is the set of edges. Furthermore, let \( w_{nm} \) be the weight of the edge between nodes \( n \) and \( m \), \( n, m \in N \). Next, denote \( c_n \) to be the community (label) of node \( n \) and \( N_n \) the set of its neighbors. Moreover, denote \( \mathcal{N}_n^l \) to be the set of neighbors of \( n \) that share label \( l \).

Label propagation algorithm (LPA) [40] reveals network communities by employing the following procedure. At first each node \( n \in N \) is labeled with an

² The work presented in this article was already (partially) presented in [46].
unique label, $c_n = l_n$. Next, at each iteration, each node adopts the label shared by most of its neighbors. Hence,

$$c_n = \arg\max_l |\mathcal{N}_n^l|,$$

(1)

where in the case of ties one of the labels is selected at random (node $n$ retains its current label, when it is among most frequent in $\mathcal{N}_n$). The process continues until none of the labels change anymore, i.e. an equilibrium is reached. During the course of the algorithm, densely connected sets of nodes form a consensus on some particular label; thus, at the end, nodes sharing the same label are classified into the same community.

Leung et al. [28] have observed that basic label propagation applied to large (web) graphs commonly produces one major community that occupies most of the nodes. However, they have shown that the emergence of a major community can be eliminated by using label hop attenuation technique. Each label $l_n$ has associated an additional score $s_n$ (initially set to 1) that decreases by $\delta$ after each propagation ($\delta$ is an attenuation ratio). When $s_n$ reaches 0, the label $l_n$ no longer propagates onward (see Eq. (4)), which successfully eliminates the emergence of a major community.

Label hop attenuation can be rewritten into an equivalent form that allows altering $\delta$ during the course of the algorithm [28]. One keeps the label distance from the origin $d_n$ (initially set to 0) that is updated after each propagation. Hence,

$$d_n = \left( \min_{m \in \mathcal{N}_n^*} d_m \right) + 1,$$

(2)

when the score $s_n$ is then

$$s_n = 1 - \delta d_n.$$

(3)

Further analysis in [28] has revealed that label hop attenuation has to be coupled with node preference $f_m$ (i.e. node propagation strength), in order for the algorithm to improve on the basic label propagation. Thus, the label propagation updating rule in Eq. (1) is transformed into

$$c_n = \arg\max_l \sum_{m \in \mathcal{N}_n^l} f_m s_m w_{nm},$$

(4)

where $\alpha$ is a parameter of the algorithm. Leung et al. [28] have experimented with node preference equal to the degree of the node (i.e. $f_m = \text{deg}_m$ and $\alpha = 0.1$), however, no general analysis was conducted.

The updating rule of label propagation (Eq. (1)), or its refinements (Eq. (4)), might prevent the algorithm from converging [40]. Imagine a bipartite network with two sets of nodes, i.e. red and blue nodes. Let, at some iteration of the algorithm, all red nodes share label $l_r$ and all blue nodes share label $l_b$. Due to the bipartite structure of the network, at the next iteration, all red, blue nodes
will adopt label $l_b$, $l_r$ respectively. Furthermore, after the next iteration, all nodes will recover their original labels, failing the algorithm to converge.

The problem can be avoided by using asynchronous updating [40]. Nodes are no longer updated all together, but sequentially, in a random order. Thus, when node’s label is updated, (possibly) already updated labels of its neighbors are considered (in contrast to synchronous updating that considers only labels from the previous iteration). All of the algorithms, presented in the following section, use such asynchronous updating of nodes.

3 Defensive and offensive label propagation

In this section we present different algorithms that employ two unique strategies of community formation, namely, defensive preservation and offensive expansion of communities. First, we briefly present a dynamic hop attenuation technique in section 3.1. Next, section 3.2 introduces and formally discusses the two strategies and associated algorithms (denoted $dDaLPA$ and $oDaLPA$ respectively). Last, section 3.3 presents an advanced label propagation algorithm (denoted $K$-Cores) that combines the two strategies in an iterative manner, thus retaining the advantages of both approaches.

3.1 Dynamic hop attenuation

Label hop attenuation has proven to be a reliable technique for prevention of emergence of a major community (section 2). Still, it is not immediately evident what should the value of attenuation ratio $\delta$ be. In [28] authors have obtained good results with values around 0.1, however, only a limited set of networks was considered.

We propose a dynamic hop attenuation technique based on the hypothesis that hop attenuation should only be employed when a label, or a set of labels, rapidly occupies a large portion of the network (which could potentially result in a formation of a major community). Otherwise, the restriction should be (almost) completely relaxed to allow label propagation to reach the equilibrium unrestrained. The technique would thus retain the dynamics of label propagation, but still successfully prevent the emergence of a major community.

We employ the following hop attenuation strategy. After each iteration (i.e. sweep through all the nodes) $\delta$ is set to the proportion of nodes that changed their labels (on the first two iterations $\delta$ is set to 0.5 and 0.1 respectively). In practice, this results in higher values of $\delta$ in the early iterations of the algorithm, which enables the occurrence of a larger number of (smaller) well defined communities, when in the later stages $\delta$ gradually converges to 0, which refines the communities and preserves only those strongly depicted in the network topology. Moreover, empirical analysis on real-world networks shows that such dynamic strategy

---

3 Similar idea was already discussed in [28].
successfully eliminates the emergence of a major community (the exact results are omitted).

Note that an additional constraint should be imposed to prevent extremely large values of \( \delta \) in the late stages of the algorithm (which, due to the above discussion, indicates some spurious behavior). Thus, to ensure convergence, when \( \delta \) is greater than \( \delta_{\text{max}} \) (e.g. \( \delta_{\text{max}} = 0.5 \)), we set it to 0.

3.2 Defensive preservation and offensive expansion of communities

Leung et al. [28] have shown that applying node preference (section 2), to alter propagation strength or spread from certain nodes, can greatly improve the performance of the basic label propagation. Nevertheless, our empirical analysis has revealed that different networks favor different strategies for node preference [46].

On small social networks, where high degree nodes reside in the core of each community (e.g. Zachary’s karate club network [50]), good performance can be obtained by using degree or eigenvector centrality [12, 13] for node preference. However, on Girvan and Newman [14] benchmark networks with planted partition, where all nodes have equal degree (on average), the measures render useless and are outperformed by clustering coefficient [48]. Furthermore, on Lancichinetti et al. [22] benchmark networks superior performance is obtained by using inverted degree or inverted eigenvector centrality. Interestingly, the measures thus complement each node’s degree, decreasing the propagation strength from high degree nodes (and vice-versa). In summary, the analysis has revealed that none of the considered measures is appropriate for general networks (all different kinds of networks).

We have observed that, during the course of the algorithm, applying node preference to the core of each current community (i.e. to its most central nodes) can significantly increase the performance on a wide range of real-world networks. Furthermore, the strategy results in a great ability of detecting communities, even when they are only weakly defined. On the other hand, applying node preference to the border of each current community (i.e. to its edge nodes) results in an extremely accurate detection, expanding communities that are strongly depicted in the network topology.

Based on above observations we propose two algorithms that convey two unique strategies of community formation. The algorithms estimate the core (and border) of each (current) community by means of the diffusion over the network; and are denoted defensive and offensive diffusion label propagation algorithm (dDaLPA and oDaLPA respectively). Let \( p_n \in (0, 1) \) be a value for node \( n \in N \) thus that nodes in the core of the community have higher values of \( p_n \) than border nodes. The defensive algorithm dDaLPA applies preference (i.e. propagation strength) to the core of each community, i.e. \( f_n = p_n \), and the

\[\frac{\text{The proportion of nodes that change their labels on the first five iterations roughly follows the sequence 90\%, 30\%, 10\%, 5\%, 3\% [46] (on networks of moderate size).}}{\text{91}}\]
Fig. 1. Comparison of defensive and offensive label propagation on two real-world networks (see Table 1). Revealed communities are shown with pentagonal nodes, when the sizes (and colors) of nodes are proportional to the sizes of communities. Defensive propagation produces a larger set of communities that are (on average) considerably smaller than those revealed by the offensive propagation.

The updating rule in Eq. (4) rewrites to

\[ c_n = \arg\max_l \sum_{m \in N_n^l} p_m s_m w_{nm}. \tag{5} \]

On the other hand, the offensive version \textit{oDaLPA} applies preference to the border of each community, i.e. \( f_n^a = 1 - p_n \), and the updating rule becomes

\[ c_n = \arg\max_l \sum_{m \in N_n^l} (1 - p_m) s_m w_{nm}. \tag{6} \]

During the course of the algorithm, values \( p_n \) are estimated using random walks within each current community. Let \( p_n \) be the probability that a random walker, utilized on the community labeled with \( c_n \), visits node \( n \) (due to simplicity, we assume that community features connectedness). \( p_n \) can then be computed as

\[ p_n = \sum_{m \in N_n^a} p_m / \deg^{c_n}_m. \tag{7} \]

where \( \deg^{c_n}_m \) is the intra-community degree of node \( m \) (\( c_m = c_n \)). Besides deriving an estimate of the core and border of each community, the rationale here is to formulate label propagation (i.e. diffusion) within each of the current communities. Thus, opposed to the algorithm in [28], the main novelty is in considering (current) communities, found by the algorithm, to estimate the (current) state of the label propagation process and then to adequately alter the dynamics of the process.

Defensive and offensive label propagation algorithms result in two unique strategies of community formation, namely, \textit{defensive preservation} and \textit{offensive preservation}. 

92
expansion of communities. The defensive algorithm quickly establishes a larger number of strong community cores (in the sense of Eq. (5)) and is able to defensively preserve them during the course of the algorithm. This results in an immense ability of detecting communities, even when they are only weakly defined in the network topology. On the other hand, the offensive approach produces a much smaller set of communities (of various sizes). Laying the pressure on the edge of each community expands (i.e. enlarges) those that are strongly depicted in the network topology. This constitutes a more natural (offensive) struggle among communities and results in a great accuracy of the communities revealed.

Comparison of the approaches on two real-world networks is shown in Fig. 1; and for pseudo-code of the algorithms see Alg. 1.

**Algorithm 1** Defensive label propagation algorithm (dDaLPA).

**Input:** Undirected graph $G(N, E)$ with weights $W$  

**Output:** Communities $C$ (i.e. node labels) 

$\delta \leftarrow 0.5$  

for $n \in N$ do  

$c_n \leftarrow l_n \{\text{Unique label.}\}$  

$p_n \leftarrow 1/|N|$  

$d_n \leftarrow 0$  

end for  

while not converged do  

shuffle($N$)  

for $n \in N$ do  

$c_n \leftarrow \arg \max c\in C \sum_{m \in N_n} p_m (1 - \delta d_m) w_{nm} \{1 - p_m \text{ instead of } p_n \text{ for oDaLPA.}\}$  

$p_n \leftarrow \sum_{m \in N_n} p_m / \deg_m^n \{\text{deg}_m^n \text{ instead of } \deg_m^n \text{ for oDaLPA.}\}$  

if $c_n \text{ has changed}$ then  

$d_n \leftarrow (\min_{m \in N_n} d_m) + 1$  

end if  

end for  

$\delta \leftarrow \text{proportion of labels changed} \{\delta \leftarrow 0.1 \text{ at the first iteration.}\}$  

if $\delta \geq \delta_{\text{max}}$ then  

$\delta \leftarrow 0 \{\text{Ensuring convergence (} \delta_{\text{max}} \text{ is fixed to 0.5).}\}$  

end if  

end while  

return $C$ \{Returns (relabeled) communities that feature connectedness.\}

3.3 Combining defensive and offensive propagation

Defensive and offensive label propagation (section 3.2) convey two unique strategies of community formation. An obvious improvement would be to combine the strategies, thus retaining the strong detection ability of the defensive approach and high accuracy of the offensive strategy. However, simply using the algorithms one after another does not attain the desired properties – any label propagation
algorithm, being run until convergence, finds a local optimum (i.e. local equilibrium) that is hard to escape from.

Raghavan et al. [40] have already discussed the idea (however, in different context) that label propagation could be improved, if one had a priori knowledge about community cores. Core nodes could then be labeled with the same label, leaving all the other nodes labeled with an unique label. During the course of the algorithm, the (uniquely labeled) nodes would tend to adopt the label of their nearest attractor (i.e. community core) and thus join its community. This would improve the algorithm’s stability [40] and also the accuracy of the identified communities (section 4).

The defensive and offensive label propagation algorithms are thus combined in the following manner (Fig. 2). First, the defensive strategy is applied, to produce initial estimates of the communities and to accurately detect their cores. All border nodes of each community are then relabeled (labeled with unique labels), so that (approximately) one half of the nodes retain their original label. Next, the offensive strategy is applied, which refines the community cores and accurately detects also their borders. Relabeling and offensive refinement are then repeated until the number of communities decreases. Such combined strategy preserves advantages of both, defensive and offensive, label propagation (section 4) and is denoted K-Cores\(^5\) algorithm (due to its resemblance to a well-known K-Means algorithm [31]).

The core (and border) of each community is again estimated by means of (diffusion) values \(p_n\) (section 3.2). Thus, within the algorithm, the node \(n\) is

\(^5\) The term should not be confused with \(k\)-core [44] that denotes the maximal subgraph in which each node has degree at least \(k\).
relabeled due to the following rule,

\[ c_n = \begin{cases} c_n & \text{for } p_n > m_{cn} \\ l_n & \text{for } p_n \leq m_{cn} \end{cases} \]  

(8a) \hspace{1cm} (8b)

where \( m_{cn} \) is the median of values \( p_n \), for nodes in the community \( c_n \), and \( l_n \) is an unique label. Hence, the core nodes retain their original labels, when all border nodes are relabeled.

Schematic representation of the algorithm is depicted in Fig. 2; and for the pseudo-code of the algorithm see Alg. 2. For a further discussion on all presented algorithms see [46].

**Algorithm 2 K-Cores algorithm.**

**Input:** Undirected graph \( G(N,E) \) with weights \( W \)

**Output:** Communities \( C \) (i.e. node labels)

\[ C \leftarrow \text{dDaLPA}(G,W) \{ \text{Defensive label propagation.} \} \]

while \( |C| \) decreases do

for \( c \in C \) do

\[ m_c \leftarrow \text{median}\{p_n \mid n \in N \land c_n = c\} \{ \text{Retain community cores.} \} \]

for \( n \in N \) and \( c_n = c \) do

if \( p_n \leq m_c \) then

\[ c_n \leftarrow l_n \{ \text{Unique label.} \} \]

\[ p_n \leftarrow 1/|N| \]

end if

\[ d_n \leftarrow 0 \]

end for

end for

\[ C \leftarrow \text{oDaLPA}(G,W) \{ \text{Offensive label propagation.} \} \]

end while

**return** \( C \{ \text{Returns best communities found.} \} \)

**4 Empirical evaluation and discussion**

In this section we present and discuss results of the empirical evaluation of the proposed algorithms. Section 4.1 gives results of the analysis on benchmark networks with planted partition, when the results on real-world network are reported in section 4.2. For the use with larger networks, we also briefly present and empirically compare two manners of hierarchical community detection in section 4.3.

The results are assessed using two measures of community structure, namely, *Normalized Mutual Information NMI* [8] and *modularity Q* [35]. The latter measures the relative significance of the communities due to a selected null model. Let \( A_{nm} \) denote the number of edges incident to nodes \( n, m \in N \) and let \( P_{nm} \) be the expected number of incident edges in the null model. The modularity then
reads
\[
Q = \frac{1}{2|E|} \sum_{n,m \in N} (A_{nm} - P_{nm}) \delta(c_n, c_m),
\]
where \(c_n\) is the identified community (label) for node \(n \in N\) and \(\delta\) is the Kronecker delta. The modularity thus measures the fraction of the difference between the number intra-community edges and the expected number of edges in the null model \((Q \in [-1, 1])\). Commonly a random graph with the same degree distribution as the original is selected for the null model. Hence, \(P_{nm} = \frac{\text{deg}_n \cdot \text{deg}_m}{2|E|}\).

Furthermore, the analysis on networks with planted partition is conducted using Normalized Mutual Information NMI \([8]\). Let \(C\) be the partition (i.e. communities) extracted by some algorithm and let \(P\) be the planted partition of the network (corresponding random variables are \(C\) and \(P\) respectively). The NMI of \(C\) and \(P\) is then
\[
\text{NMI} = \frac{2I(C, P)}{H(C) + H(P)},
\]
where \(I(C, P)\) is the mutual information of the partitions, \(I(C, P) = H(C) - H(C|P)\), and \(H(C)\), \(H(P)\) and \(H(C|P)\) are standard and conditional entropies. NMI of identical partitions equals 1, and is 0 for independent partitions.

### 4.1 Networks with planted partition

The proposed algorithms were first analyzed on four different types of Lancichinetti et al. \([22]\) benchmark networks with planted partition. The results are shown in Fig. 3.

The analysis clearly depicts the difference between defensive and offensive label propagation. The offensive approach (\(oDaLPA\)) performs considerably better than the basic label propagation (\(LPA\)) and can still accurately detect communities, when \(LPA\) already fails. On the other hand, the defensive propagation (\(dDaLPA\)) does not detect communities as accurately as the offensive approach, and \(LPA\) on larger networks, but still reveals communities, even when they are only weakly defined. Furthermore, \(K\)-Cores algorithm outperforms all other approaches in all but one case. Note that the algorithm retains the advantages of both defensive and offensive approach, still, the performance does not simply equal to the upper-hull of those for \(dDaLPA\) and \(oDaLPA\).

The algorithms were also applied to a random graph à la Erdős-Rényi \([10]\) that (presumably) has no community structure (Fig. 4). However, the defensive label propagation still reports communities, when the average degree is small enough. Nevertheless, further analysis reveals that defensive label propagation is still a preferred approach on a wide range of real-world networks (section 4.2).

As \(K\)-Cores algorithm is initialized using the defensive propagation, and best communities are reported at the end, the performance for \(K\)-Cores on a random graph is similar to that for \(dDaLPA\). However, if we discard the initial communities obtained by \(dDaLPA\) (i.e. \(K\)-Cores\(^4\) algorithm), the results correspond to those for \(LPA\) and \(oDaLPA\) that reveal no community structure.
Fig. 3. Comparison of the proposed algorithms on Lancichinetti et al. [22] benchmarks networks with planted partition. The network sizes equal 1000 and 5000 nodes respectively; and communities comprise of up to 50 and 100 nodes respectively. The results were averaged over 100 realizations of the benchmarks networks.

Fig. 4. Comparison of the proposed algorithms on a random graph à la Erdös-Rényi [10] with 1000 nodes (the results were averaged over 10 runs).
4.2 Real-world networks

The algorithms were further analyzed on over 20 real-world networks of moderate size (Table 2). Due to a large number of networks considered, the detailed description is omitted (see Table 1). However, the set includes different communication, social, biological, web, (author) collaboration, Internet and other networks. Due to simplicity, all networks are considered as unweighted and undirected, i.e. all weighted or directed edges are treated as simple undirected edges (same holds for networks in section 4.3).

We also introduce a new type of networks denoted software networks (sort of component dependency networks [49]). Here nodes represent a set of classes of some software system, written in an object-oriented programming language, and edges represent relations among them. Two classes $A$ and $B$ are defined as related when $B$ extends or implements $A$, when $B$ contains a field of type $A$ or when $A$, $B$ contains a method that returns, requires an object of type $B$, $A$ respectively. The hypothesis here is that network communities would correspond to software packages, which could result in numerous applications in software engineering domain (Fig. 5). In this article we consider the ground case, where networks are represented with simple undirected and unweighted graphs$^6$.

---

Fig. 5. Communities revealed in javax software network by applying $K$-Cores algorithm. The sizes of nodes correspond to the sizes of communities; and the widths of the edges correspond to the number of inter-community edges (due to clarity, weakly represented nodes and edges were discarded). Text shows the distribution of javax packages within the communities, where all weakly represented packages were omitted.

---

$^6$ The networks were obtained by parsing the documentation of the corresponding software. Thus, due to various reasons, some false relations might have been introduced.
Table 1. Networks used for the analysis of community detection algorithms.

<table>
<thead>
<tr>
<th>Network</th>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>uni</td>
<td>Emails within an university.</td>
<td>[17]</td>
</tr>
<tr>
<td>enron</td>
<td>Emails within Enron.</td>
<td>[25]</td>
</tr>
<tr>
<td>football</td>
<td>American college football league.</td>
<td>[14]</td>
</tr>
<tr>
<td>jazz</td>
<td>Network of jazz musicians.</td>
<td>[15]</td>
</tr>
<tr>
<td>wiki</td>
<td>Voting network of Wikipedia.</td>
<td>[24]</td>
</tr>
<tr>
<td>epinions</td>
<td>Epinions web of trust.</td>
<td>[41]</td>
</tr>
<tr>
<td>yeast</td>
<td>Yeast protein interactions.</td>
<td>[20]</td>
</tr>
<tr>
<td>elegans</td>
<td>Nematode Caenorhabditis elegans.</td>
<td>[21]</td>
</tr>
<tr>
<td>gnutella</td>
<td>Gnutella peer-to-peer network.</td>
<td>[26]</td>
</tr>
<tr>
<td>blogs</td>
<td>Weblogs on U.S. politics.</td>
<td>[2]</td>
</tr>
<tr>
<td>genrelat</td>
<td>General Relativity archive 2003.</td>
<td>[26]</td>
</tr>
<tr>
<td>codmat3</td>
<td>Condensed Matter archive 2003.</td>
<td>[32]</td>
</tr>
<tr>
<td>codmat5</td>
<td>Condensed Matter archive 2005.</td>
<td>[32]</td>
</tr>
<tr>
<td>hep</td>
<td>High Energy Physics archive 2003.</td>
<td>[26]</td>
</tr>
<tr>
<td>astro</td>
<td>Astro Physics archive 2003.</td>
<td>[26]</td>
</tr>
<tr>
<td>engine</td>
<td>Google App Engine library.</td>
<td></td>
</tr>
<tr>
<td>jung</td>
<td>JUNG graph and network library.</td>
<td></td>
</tr>
<tr>
<td>javax</td>
<td>Java 6 javax namespace.</td>
<td></td>
</tr>
<tr>
<td>power</td>
<td>Western U.S. power grid.</td>
<td>[48]</td>
</tr>
<tr>
<td>oregon3</td>
<td>Aut. syst. of Internet 2003 (Oregon).</td>
<td>[25]</td>
</tr>
<tr>
<td>oregon6</td>
<td>Aut. syst. of Internet 2006 (Oregon).</td>
<td>[34]</td>
</tr>
<tr>
<td>nec</td>
<td>nec web overlay map.</td>
<td>[19]</td>
</tr>
<tr>
<td>amazon</td>
<td>Amazon co-purchasing network.</td>
<td>[23]</td>
</tr>
<tr>
<td>ndedu</td>
<td>Web graph of nd.edu domain.</td>
<td>[3]</td>
</tr>
<tr>
<td>road</td>
<td>Roads in Pennsylvania.</td>
<td>[27]</td>
</tr>
<tr>
<td>google</td>
<td>Web graph of Google.</td>
<td>[27]</td>
</tr>
<tr>
<td>skitter</td>
<td>Aut. syst. of Internet 2005 (Skitter).</td>
<td>[25]</td>
</tr>
<tr>
<td>move</td>
<td>Movie actors collaborations.</td>
<td>[4]</td>
</tr>
<tr>
<td>nber</td>
<td>NBER patents citations.</td>
<td>[18]</td>
</tr>
<tr>
<td>live</td>
<td>Live Journal friendships.</td>
<td>[27]</td>
</tr>
<tr>
<td>webbase</td>
<td>Web graph from WebBase.</td>
<td>[1]</td>
</tr>
</tbody>
</table>
## Table 2. Mean modularities \( Q \) for different label propagation algorithms (averaged over 100 to 100000 runs). The highest values of \( Q \) are shown with solid font; and underlined values correspond to the highest values among only \( dDaLPA \) and \( oDaLPA \).

<table>
<thead>
<tr>
<th>Type</th>
<th>Network</th>
<th>Nodes</th>
<th>Edges</th>
<th>LPA</th>
<th>( dDaLPA )</th>
<th>( oDaLPA )</th>
<th>( K)-Cores</th>
</tr>
</thead>
<tbody>
<tr>
<td>Communication</td>
<td>uni</td>
<td>1133</td>
<td>5451</td>
<td>0.364</td>
<td>0.481</td>
<td>0.389</td>
<td>0.518</td>
</tr>
<tr>
<td></td>
<td>enron</td>
<td>36692</td>
<td>367662</td>
<td>0.355</td>
<td>0.514</td>
<td>0.380</td>
<td>0.516</td>
</tr>
<tr>
<td>Social</td>
<td>football</td>
<td>115</td>
<td>616</td>
<td>0.592</td>
<td>0.593</td>
<td>0.595</td>
<td>0.600</td>
</tr>
<tr>
<td></td>
<td>jazz</td>
<td>198</td>
<td>2742</td>
<td>0.346</td>
<td>0.418</td>
<td>0.377</td>
<td>0.418</td>
</tr>
<tr>
<td></td>
<td>wiki</td>
<td>7115</td>
<td>103689</td>
<td>0.056</td>
<td>0.195</td>
<td>0.046</td>
<td>0.202</td>
</tr>
<tr>
<td></td>
<td>epinions</td>
<td>75879</td>
<td>508837</td>
<td>0.106</td>
<td>0.288</td>
<td>0.111</td>
<td>0.291</td>
</tr>
<tr>
<td>Protein</td>
<td>yeast</td>
<td>2114</td>
<td>4480</td>
<td>0.665</td>
<td>0.733</td>
<td>0.729</td>
<td>0.793</td>
</tr>
<tr>
<td>Metabolic</td>
<td>elegans</td>
<td>453</td>
<td>2025</td>
<td>0.122</td>
<td>0.172</td>
<td>0.131</td>
<td>0.173</td>
</tr>
<tr>
<td>Peer-to-peer</td>
<td>gnutella</td>
<td>62586</td>
<td>147892</td>
<td>0.338</td>
<td>0.412</td>
<td>0.387</td>
<td>0.447</td>
</tr>
<tr>
<td>Web</td>
<td>blogs</td>
<td>1490</td>
<td>16718</td>
<td>0.400</td>
<td>0.424</td>
<td>0.424</td>
<td>0.426</td>
</tr>
<tr>
<td>Collaboration</td>
<td>genrelat</td>
<td>5242</td>
<td>28980</td>
<td>0.737</td>
<td>0.769</td>
<td>0.779</td>
<td>0.820</td>
</tr>
<tr>
<td></td>
<td>codmat3</td>
<td>27519</td>
<td>116181</td>
<td>0.596</td>
<td>0.611</td>
<td>0.627</td>
<td>0.687</td>
</tr>
<tr>
<td></td>
<td>codmat5</td>
<td>36458</td>
<td>171736</td>
<td>0.548</td>
<td>0.575</td>
<td>0.590</td>
<td>0.648</td>
</tr>
<tr>
<td></td>
<td>hep</td>
<td>12008</td>
<td>237010</td>
<td>0.484</td>
<td>\textbf{0.585}</td>
<td>0.518</td>
<td>0.585</td>
</tr>
<tr>
<td></td>
<td>astro</td>
<td>18772</td>
<td>396160</td>
<td>0.326</td>
<td>\underline{0.538}</td>
<td>0.337</td>
<td>0.538</td>
</tr>
<tr>
<td>Software</td>
<td>engine</td>
<td>139</td>
<td>243</td>
<td>0.689</td>
<td>0.724</td>
<td>0.726</td>
<td>0.747</td>
</tr>
<tr>
<td></td>
<td>jung</td>
<td>436</td>
<td>1303</td>
<td>0.611</td>
<td>0.587</td>
<td>0.623</td>
<td>0.631</td>
</tr>
<tr>
<td></td>
<td>javax</td>
<td>2089</td>
<td>7934</td>
<td>0.723</td>
<td>0.687</td>
<td>0.725</td>
<td>0.768</td>
</tr>
<tr>
<td>Power</td>
<td>power</td>
<td>4941</td>
<td>6594</td>
<td>0.595</td>
<td>0.690</td>
<td>0.698</td>
<td>0.820</td>
</tr>
<tr>
<td>Internet</td>
<td>oregon3</td>
<td>767</td>
<td>3591</td>
<td>0.302</td>
<td>0.210</td>
<td>\textbf{0.354}</td>
<td>0.210</td>
</tr>
<tr>
<td></td>
<td>oregon6</td>
<td>22963</td>
<td>48436</td>
<td>0.498</td>
<td>0.347</td>
<td>\underline{0.541}</td>
<td>0.347</td>
</tr>
<tr>
<td></td>
<td>nec</td>
<td>75885</td>
<td>357317</td>
<td>0.683</td>
<td>0.628</td>
<td>0.688</td>
<td>\underline{0.736}</td>
</tr>
</tbody>
</table>

Comparison of the algorithms on real-world networks (Table 2) firstly confirms the adequacy of the \( K\)-Cores algorithm that obtains highest modularity on all but two Internet networks. Note that both \( dDaLPA \) and \( oDaLPA \) also outperform the basic \( LPA \) in most cases. Moreover, the analysis clearly separates different types of networks due to the preferred strategy of community formation. For instance, social and communication networks clearly favor defensive preservation of communities, due to high density of such networks (and rather weakly defined communities). On the other hand, sparse software or Internet networks, with longer paths among nodes, obviously prefer the offensive expansion of communities. The middle case is represented by the considered collaboration networks. On smaller networks that are relatively sparse (\textit{genrelat}, \textit{codmat}3 and \textit{codmat}5 network) the offensive approach prevails. However, on larger networks (\textit{hep} and \textit{astro} network), with significantly higher degrees then the former, the defensive algorithm is superior. In summary, denser networks (with higher av-
verage degrees) prefer the defensive preservation, whereas sparser networks (with lower average degrees) favor the offensive expansion of communities.

4.3 Analyzing large networks

Last, we also briefly present and empirically evaluate two different manners of hierarchical community investigation (prominent for the use with larger networks). Besides LPA and K-Cores algorithm, we consider the following approaches.

Basic diffusion and propagation algorithm \( \text{(DPA)} \) \cite{46} is an optimized version of K-Cores that scales significantly better than the basic algorithm. Furthermore, hierarchical diffusion and propagation algorithm \( \text{(DPA}^+ \text{)} \) represents a simple hierarchical detection, where the algorithm is recursively applied to the previously constructed community network \(^7\) (the algorithm employs defensive label propagation, and DPA on the last step). Moreover, (general) diffusion and propagation algorithm \( \text{(DPA}^* \text{)} \) \cite{46} represents a hierarchical core extraction technique, where the algorithm recursively extracts the core \cite{27} of the network and identifies whisker communities. For a further discussion on the algorithms see \cite{46}.

Table 3. Peak modularities \( Q \) and average number of iterations for different label propagation algorithms (obtained over 1 to 10 runs). Solid values correspond to the largest values of \( Q \), where missing values could not be obtained due to limited time resources.

<table>
<thead>
<tr>
<th>Network</th>
<th>Nodes</th>
<th>Edges</th>
<th>LPA</th>
<th>K-Cores</th>
<th>DPA</th>
<th>DPA(^+)</th>
<th>DPA(^*)</th>
</tr>
</thead>
<tbody>
<tr>
<td>amazon</td>
<td>0.3M</td>
<td>1.2M</td>
<td>0.681/15</td>
<td>0.783/273</td>
<td>0.700/34</td>
<td><strong>0.883/65</strong></td>
<td>0.856/78</td>
</tr>
<tr>
<td>ndedu</td>
<td>0.3M</td>
<td>1.5M</td>
<td>0.838/53</td>
<td>0.891/471</td>
<td>0.860/50</td>
<td>0.897/37</td>
<td><strong>0.901/58</strong></td>
</tr>
<tr>
<td>road</td>
<td>1.1M</td>
<td>3.1M</td>
<td>0.552/10</td>
<td>0.847/895</td>
<td>0.626/82</td>
<td><strong>0.985/136</strong></td>
<td>0.883/142</td>
</tr>
<tr>
<td>google</td>
<td>0.9M</td>
<td>4.3M</td>
<td>0.801/15</td>
<td>0.889/444</td>
<td>0.820/59</td>
<td>0.962/45</td>
<td><strong>0.967/48</strong></td>
</tr>
<tr>
<td>skitter</td>
<td>1.7M</td>
<td>11.1M</td>
<td>0.746/25</td>
<td>-</td>
<td>0.755/126</td>
<td>0.680/52</td>
<td><strong>0.801/76</strong></td>
</tr>
<tr>
<td>movie</td>
<td>0.4M</td>
<td>15.0M</td>
<td>0.524/21</td>
<td>-</td>
<td>0.533/147</td>
<td>0.474/39</td>
<td><strong>0.606/71</strong></td>
</tr>
<tr>
<td>nber</td>
<td>3.8M</td>
<td>16.5M</td>
<td>0.576/109</td>
<td>-</td>
<td>0.582/336</td>
<td>0.707/112</td>
<td><strong>0.739/308</strong></td>
</tr>
<tr>
<td>live</td>
<td>4.8M</td>
<td>69.0M</td>
<td>0.673/100</td>
<td>-</td>
<td>0.548/206</td>
<td>0.683/73</td>
<td><strong>0.688/125</strong></td>
</tr>
<tr>
<td>webbase</td>
<td>14.5M</td>
<td>101.0M</td>
<td>0.894/38</td>
<td>-</td>
<td>0.923/114</td>
<td>0.942/43</td>
<td><strong>0.954/39</strong></td>
</tr>
</tbody>
</table>

Algorithms were applied to a set of large real-world networks \( \text{(Table 3)} \) of various types (see Table 1), when the analysis on (even) larger networks was limited due to limited memory resources \( \text{(i.e. 4 GB of memory)} \). On average, all of the considered algorithms again perform better than the basic label propagation \( \text{(LPA)} \). Furthermore, the hierarchical algorithms \( \text{(DPA}^+ \text{ and DPA}^* \text{)} \) obtain the highest values of modularity on all of the networks considered, whereas the core extraction technique \( \text{(DPA}^* \text{)} \) seems more prominent.

\(^7\) A network whose nodes represent communities and edges represent edges between nodes in the original network.
The average number of iterations made by the algorithms\(^8\) (Table 3) shows that “theoretical” approach K-Cores does not scale to larger networks, where the optimized version DPA is preferred. Note also that hierarchical detection even decreases the total number of iterations on larger networks, which gives promising grounds for future analysis of large complex networks.

For a further empirical evaluation and comparison with other label propagation algorithms reported in the literature see [46].

5 Conclusion

In the article we present different label propagation algorithms that employ two unique strategies of community formation, namely, defensive preservation and offensive expansion of communities. The strategies are combined in an advanced label propagation algorithm that retains the advantages of both approaches. Furthermore, we also show how the algorithm can be extended to larger networks using (hierarchical) core extraction. Nevertheless, the main contribution of this work is in showing that different types of networks (with different topological properties) favor different strategies of community formation.

Future work will focus mainly on further analyses of defensive and offensive label propagation, in order to develop an enhanced algorithm that would decide between defensive and offensive strategy (an intermediate approaches) during the course of the algorithm. This could result in higher accuracy of the revealed communities and also in better scalability of the algorithm.

References


\(^8\) As the employed implementation was optimized for memory, not time, resources, we report the number of iterations instead of the exact running times.
Applications of stochastic relational processes *

Ingo Thon¹, Niels Landwehr², and Luc De Raedt¹

¹Department of Computer Science, Katholieke Universiteit Leuven, Celestijnenlaan 200A, 3001 Heverlee, Belgium
²Department of Computer Science, University of Potsdam, August-Bebel-Str. 89, 14482 Potsdam, Germany
ingo.thon@cs.kuleuven.be, landwehr@cs.uni-potsdam.de, luc.deraedt@cs.kuleuven.be

Abstract. Real world data collections typically feature a variable number of objects, relations amongst them, and non-deterministic transition behavior. Traditional graph miner extracting rules from such data collections are not able to quantify the non-determinism. While standard probabilistic sequence models provide efficient inference and learning techniques for sequential data, they typically cannot fully capture the relational complexity. On the other hand, statistical relational learning techniques are often too inefficient to cope with complex sequential data. We introduced a simple model that occupies an intermediate position in this expressiveness/efficiency trade-off. It is based on CP-logic (Causal Probabilistic Logic), an expressive probabilistic logic for modeling causality. However, by specializing CP-logic to represent a probability distribution over sequences of relational state descriptions and employing a Markov assumption, inference and learning become more tractable and effective.

We experimentally validate that the resulting technique is able to handle probabilistic relational domains with a substantial number of objects and relations.

1 Introduction

Many real-world domains are best described as a large graph. While graphs are a proper description of the current state of the world, they remain a snapshot, absorbing the information about the history. To properly account for world dynamics, domains such rather be described by a sequence of graphs, where each element describes the world state at a given point in time. In this paper, we discuss a generative probabilistic model for sequences of labeled (hyper)graphs, cf. Figure 3 (2-4) for an example. Such a model represents a stochastic process with complex state representations, and thereby covers a wide range of application domains. It is appropriate whenever systems evolve over time and are complex enough that their states cannot easily be described using a propositional representation. A prominent example are states that are characterized by a graph structure relating different agents and/or world artifacts at a given point in time (as in dynamic social networks, computer networks, the world wide web, games, marketplaces, et cetera). Due to technological developments in processing, sharing, and

* This is a condensed version of a paper accepted for publication in the Machine Learning Journal special issue on Mining and Learning with Graphs and Relations.
storing data (in particular in the context of the world wide web) the availability and the
size of such datasets is rapidly increasing.

As a concrete running example, we shall focus in this paper on modeling massively
multiplayer online games (MMOGs) These are computer games that support thousands
of players in complex, persistent, and dynamic virtual worlds. They form an ideal and
realistic testbed for developing and evaluating artificial intelligence techniques and are
also interesting in their own right (cf. also [1]). One challenge in such games is to build
a dynamic probabilistic model of high-level player behavior, such as players joining or
leaving alliances and concerted actions by players within one alliance. Such a model of
human cooperative behavior can be useful in several ways. Analysis of in-game social
networks is not only interesting from a sociological point of view but could also be used
to visualize aspects of the gaming environment or give advice to inexperienced players
e.g., which alliance to join). The model could also serve to detect non-human players
in today’s MMOGs—accounts which are played by automatic scripts to give one player
an unfair advantage, and are typically against game rules.

Dynamically evolving graphs have mainly been investigated from two research per-
spectives: Graph Mining and Statistical Relational Learning (SRL). Most research con-
cerned with evolving graphs within the graph mining community has focused on the
study of global parameters (however, see [2] for a notable exception). Understanding
the evolution of graphs at a microscopic level is often more challenging but can be
equally important. For instance, identifying rule violations in online games require us
to analyze the (local) behavior of a single player; similarly, marketing departments are
interested in identifying (local) “opinion leaders” or “influencers” in a given social net-
work.

From a machine learning perspective, the outlined type of domain poses three main
challenges: (1) world state descriptions are inherently relational (namely, graphs con-
sisting of nodes and edges); (2) the transition behavior of the world is strongly stochas-
tic; and (3) a relatively large number of objects and relations is needed to build mean-
ingful models.

The field of SRL has contributed various formalisms integrating relational represen-
tations with methods for reasoning about uncertainty (for instance, Markov Logic [3],
CP-logic [4], or Bayesian Logic Programs [5]). However, inference and learning of-
ten causes significant computational problems in realistic applications, and hence such
methods do not satisfy requirement (3). To alleviate this situation, we have recently
contributed a novel representation called CPT-L (for Causal Probabilistic Time-Logic)
that occupies an intermediate position in this expressiveness/efficiency trade-off. CPT-L
defines a probability distribution over sequences of relational states but maintains effi-
ciency by focusing on the sequential aspect and deliberately avoiding the complications
that arise when dealing with hidden variables.

Outline: We proceed by first explaining the semantics of CPT-L. We then define the
inference tasks which can be addressed using our system. Afterwards, we empirically
demonstrate the system’s usefulness in two domains. Finally we conclude and touch
upon related work. For details on the algorithms please refer to [6] and [7].
Let us first introduce some terminology. A logical \textit{atom} is an expression of the form \( p(t_1, \ldots, t_n) \) where \( p/n \) is a \textit{predicate symbol} and the \( t_i \) are \textit{terms}. Terms are built up from constants, variables, and functor symbols. Constants are denoted in lower case (such as \( a \)), variables in upper case (such as \( Z \)), and functors by \( f/k \) where \( k \) is the arity of functor \( f \). The set of all atoms is called a \textit{language} \( L \). Ground expressions do not contain variables. Ground atoms will be called \textit{facts}. A \textit{substitution} \( \theta \) is a mapping from variables to terms, and \( b\theta \) is the atom obtained from \( b \) by replacing variables with terms according to \( \theta \). As an example, consider the substitution \( \theta = \{ Z/a \} \) that replaces variable \( Z \) with \( a \), as in \( b\theta = p(a) \) for \( b = p(Z) \).

Complex world states can now be described in terms of \textit{interpretations}. An interpretation \( I \) is a set of ground facts \( \{a_1, \ldots, a_N\} \). These ground facts can represent objects in the current world state, their properties, and any relationship between objects. As an example, consider the representation of the state of a multiplayer game in terms of an interpretation as depicted in Figure 1. The semantics of CPT-L is based on CP-logic, a probabilistic first-order logic that defines probability distributions over interpretations [4]. CP-logic has a strong focus on causality and constructive processes: an interpretation is incrementally constructed by a process that adds facts which are probabilistic outcomes of other already given facts (the \textit{causes}). CPT-L combines the semantics of CP-logic with that of (first-order) Markov processes. Causal influences only stretch from \( I_t \) to \( I_{t+1} \) (Markov assumption), are identical for all time steps (stationarity), and all causes and outcomes are observable.

\textbf{Definition 1.} A \textit{CPT-theory/model} is a set of rules of the form
\[
r = (h_{1,1} \land \ldots \land h_{1,k_1} : p_1) \lor \ldots \lor (h_{n,1} \land \ldots \land h_{1,k_n} : p_n) \leftarrow b_1, \ldots, b_m
\]
where the \( h_{i,j} \) are logical atoms, \( p_i \in [0, 1] \) are probabilities s.t. \( \sum_{i=1}^n p_i = 1 \), and the \( b_i \) are literals (i.e., atoms or their negation).
A conjunction $h_{i,1} \wedge \ldots \wedge h_{i,k}$ in head$(r)$ will also be called a head element, and its probability $p_i$ will be denoted by $P(h_{i,1} \wedge \ldots \wedge h_{i,k} \mid r)$. The meaning of a rule is that whenever $b_1 \theta, \ldots, b_m \theta$ holds for a substitution $\theta$ in the current state $I_t$, exactly one head element $h_{i,1} \theta \wedge \ldots \wedge h_{i,k} \theta$ is chosen from head$(r)$ and all its conjuncts $h_{i,j} \theta$ are added to the next state $I_{t+1}$.

**Example 1.** Consider the following CPT-theory for the *travian world* domain:

$$
\text{conq}(P,C) : 0.039 \vee \text{nil} : 0.961 \quad \quad \text{conq}(P,C'), \text{city}(C', \ldots, P'), \text{city}(C, \ldots, P)
$$

The rule encodes that a player is likely to conquer a city of a player he or she already attacked in the previous time step.

We now show how a CPT-theory defines a distribution over sequences $I_0, \ldots, I_T$ of relational interpretations. Let us first define the concept of an applicable rule $r$ in an interpretation $I_t$. Consider a CPT rule $c_1 : p_1 \vee \ldots \vee c_n : p_n \leftarrow b_1, \ldots, b_m$. Let $\theta$ denote a substitution that grounds the rule $r$, and let $r\theta$ denote the grounded rule. A rule $r$ is applicable in $I_t$ if and only if there exists a substitution $\theta$ such that body$(r)\theta = b_1 \theta, \ldots, b_m \theta$ is true in $I_t$, denoted $I_t \models b_1 \theta, \ldots, b_m \theta$. We will most often talk about ground rules that are applicable in an interpretation.

Given a CPT-Theory $T$, the set of all applicable ground rules in state $I_t$ will be denoted as $R_t$. That is, $R_t = \{ r\theta \mid r \in T, r\theta \text{ applicable in } I_t \}$. Each ground rule applicable in $I_t$ will cause one of its grounded head elements to be selected, and the resulting atoms to become true in $I_{t+1}$. A selection $\sigma$ is a mapping from applicable ground rules $R_t$ to head elements, associating each rule $r_i \in R_t$ with one of its head elements $\sigma(r_i)$. The probability of $\sigma$ is

$$
P(\sigma) = \prod_{i=1}^k P(\sigma(r_i) \mid r_i)
$$

(1)

where $P(\sigma(r_i) \mid r_i)$ is the probability associated with head element $\sigma(r_i)$ in the rule $r_i$.

A selection $\sigma$ defines which head element is selected for every rule, and thus determines a successor interpretation $I_{t+1}$, that simply consists of all atoms appearing in selected head elements. We shall say that $\sigma$ yields $I_{t+1}$ from $I_t$, denoted $I_t \xrightarrow{\sigma} I_{t+1}$, and define

$$
P(I_{t+1} \mid I_t) = \sum_{\sigma : I_t \xrightarrow{\sigma} I_{t+1}} P(\sigma).
$$

(2)

That is, the probability of a successor interpretation $I_{t+1}$ given an interpretation $I_t$ is computed by summing the probabilities of all selections yielding $I_{t+1}$ from $I_t$. Note that $P(I_{t+1} \mid I_t) = 0$ if no selection yields $I_{t+1}$.

**Example 2.** Consider the theory

$$
\begin{align*}
  r_1 & = p(X) : 0.2 \lor q(X) : 0.8 \leftarrow q(X) \\
  r_2 & = p(a) : 0.5 \lor (q(b) \land q(c)) : 0.5 \leftarrow \neg q(b) \\
  r_3 & = p(X) : 0.7 \lor \text{nil} : 0.3 \leftarrow p(X)
\end{align*}
$$

108
Starting from $I_t = \{p(a)\}$ only the rules $r_2$ and $r_3$ are applicable, so $R_t = \{r_2, r_3\{X/a\}\}$. The possible successor states $I_{t+1}$ are therefore

\[
I_{t+1}^1 = \{p(a)\} \text{ with } P(I_{t+1}^1 \mid I_t) = 0.5 \cdot 0.7 + 0.5 \cdot 0.3 = 0.5
\]

\[
I_{t+1}^2 = \{q(b), q(c)\} \text{ with } P(I_{t+1}^2 \mid I_t) = 0.5 \cdot 0.3 = 0.15
\]

\[
I_{t+1}^3 = \{p(a), q(b), q(c)\} \text{ with } P(I_{t+1}^3 \mid I_t) = 0.5 \cdot 0.7 = 0.35
\]

As for propositional Markov processes, the probability of a sequence $I_0, \ldots, I_T$ is

\[
P(I_0, \ldots, I_T) = P(I_0) \prod_{t=0}^{T} P(I_{t+1} \mid I_t). \tag{3}
\]

Intuitively, it is clear that this defines a distribution over all sequences of interpretations of length $T$ as in the propositional case. More formally, inductive application of the product rule yields the following theorem:

**Theorem 1 (Semantics of a CPT theory).** Given an initial state $I_0$, a CPT-theory defines a discrete-time stochastic process, and therefore for $T \in \mathbb{N}$ a distribution $P(I_0, \ldots, I_T)$ over sequences of interpretations of length $T$.

### 3 Inference and Parameter Estimation in CPT-L

As for other probabilistic models, we can now ask several questions about the introduced CPT-L model:

- **Sampling**: how to sample sequences of interpretations $I_0, \ldots, I_T$ from a given CPT-theory $T$ and initial interpretation $I_0$?
- **Inference**: given a CPT-theory $T$ and a sequence of interpretations $I_0, \ldots, I_T$, what is $P(I_0, \ldots, I_T \mid T)$?
- **Parameter Estimation**: given the structure of a CPT-theory $T$ and a set of sequences of interpretations, what are the maximum-likelihood parameters of $T$?
- **Prediction**: Let $T$ be a CPT-theory, $I_0, \ldots, I_t$ a sequence of interpretations, and $F$ a first-order formula that constitutes a certain property of interest. What is the probability that $F$ holds at time $t + d$, $P(I_{t+d} \models_B F \mid T, I_0, \ldots, I_t)$?

The solution to this problems and the algorithms are described in detail in [7].

### 4 Experimental Evaluation

In this section, we experimentally validate the proposed CPT-L approach in different settings in the following real-world domains:

**Chat Room Domain** This domain is concerned with the analysis of user interaction in chat rooms. We have monitored a number of IRC chat rooms in real time, and recorded

\[1\] The implementation, models and data will be made available soon at http://www.ingothon.de/
who was sending messages to whom using the PieSpy utility [8]. This results in dynamically changing graphs of user interaction, representing the social network structure among chat room participants, cf. Figure 2 (left). We learn these dynamics using separate models for different chat rooms. The resulting set of models can be used to visualize commonalities and differences in the behavior displayed in different chat rooms, thereby characterizing the underlying user communities.

**Massively Multiplayer Online Game Domain**

As a second evaluation domain introduced in [6],[7], we consider the large-scale massively multiplayer online strategy game Travian\(^2\). Game worlds feature thousands of players, game artifacts such as cities, armies, and resources, and social player interaction in alliances. Game states in Travian are complex and richly structured, and transitions between game states highly stochastic as they are determined by player actions. We have logged the state of a “live” game server over several months, recording high-level game states as visualized in Figure 3 (2-4). We address different learning tasks in the Travian domain, such as predicting player actions (prediction setting) and identifying groups of cooperating alliances (classification setting).

The goal of our experimental study is two-fold. First, we want to evaluate the effectiveness of the proposed approach. That is, we explore whether it is possible to learn dynamic stochastic models, and to solve the resulting inference, prediction, and classification tasks. Our second goal is to evaluate the scaling behavior for domains with a large number of objects and relationships.

### 4.1 Experiments in the Chat Room Domain

For our experiments in the chat room domain, we have selected the following 7 well-frequented IRC chat rooms: [football | iphone | computer | poker | math | politics]@irc.efnet.net, and travian@irc.travian.org. Each chat room was monitored using the PieSpy utility [8], generating a sequence graphs (cf. Figure 2 (left)). For each chat room, we selected the first 100 observations in a sequence.

We have hand-coded a simple CPT-theory \(T\) for this domain, which makes use of a number of graph-theoretic properties defined in the background knowledge, such as graph centrality, node degree, closeness, betweenness, and co-citation. As an example rule, consider

\[
\text{communicates}(P_1, P_2) : 0.1 \lor \text{nil} : 0.9 \leftarrow \text{cocitation}(P_1, P_2, \text{CC})
\]

encoding that two chat participants start talking to each other if there is a third participant with whom they have both talked before. For each chat room we learn the probabilistic parameters of a CPT-theory, resulting in 7 CPT-theories \(T_1, \ldots, T_7\) with the same rule structure but different parameters. Learning took about 10 seconds per theory \(T_i\). The learned CPT-theories can be seen as a probabilistic representation of the typical interaction behavior among members of that chat room. We evaluated the likelihood \(P(S_i \mid T_j)\) of each sequence \(S_i\) under the learned CPT-theory \(T_j\). This gives an indication as to how well the behavior in chat room \(i\) is explained by the model learned for chat room \(j\).

\(^2\)www.travian.com; www.traviangames.com
Fig. 2: (Left) User interaction graphs from the Chat Room Domain. Shown are two subsequent different time points during the observation sequence recorded for the irc.travian.org chat room. (Right) Plot of the likelihood $P(S_i | T_j)$ of a sequence $S_i$ (corresponding to chat room $i$) under the CPT-theory $T_j$ (learned on chat room $j$). Rows correspond to models $T_j$ and columns to sequences $S_i$. Lighter colors indicate higher likelihoods.

The result of this experiment is visualized in Figure 2. We can distinguish different clusters of chat rooms, or, equivalently, user communities. For instance, chat rooms that are concerned with recreational topics such as travian and football (as well as iphone) are clearly distinguishable from chat rooms concerned with more “serious” topics such as math and politics. Manual inspection of the learned rule parameters showed that in the “serious” chat domains the likelihood of a communication between two players mostly depends on the betweenness and degrees of the nodes involved, while in the “recreational” chats shared cocitations are more important.

4.2 Experiments in the Massively Multiplayer Online Game Domain

In Travian, players are spread over several independent game worlds, with approximately 20,000–30,000 players interacting in a single world. Travian gameplay follows a classical strategy game setup on a large grid-map. Each player starts with a single city. During the course of the game, players harvest resources from the environment, construct buildings, research technologies, or found new cities on free tiles of the map. Additionally, players can build different military.

In the following, we will take a more high-level view of the game and focus on modeling player interaction and cooperation in alliances rather than low-level game elements such as resources, troops and buildings. Alliances constitute social networks, where diplomacy is used to settle conflicts and players compete for an influential role. Figure 3 shows a (partial) high-level view, represented as a graph structure relating cities, players and alliances which we will refer to as a game graph.

The dynamics of the game are illustrated in Figure 3 (2–4). Here, three players from the red alliance launch an attack against territory currently held by the blue and yellow alliances.
Data Collection and Preprocessing The data was collected from a “live” Travian server with approximately 25,000 active players. Over a period of three months (December 2007 till February 2008), high-level data about the current state was collected once every 24 hours. The data was represented using predicates \( \text{city}(\text{City}, X, \text{Y}, \text{Size}, \text{Player}) \), \( \text{allied}(\text{Player}, \text{Alliance}) \), \( \text{conq}(\text{Player}, \text{City}) \) and \( \text{alliance} \text{change}(\text{Player}, \text{Alliance}) \). The background knowledge contained the discretized distance \( \text{distance}(C_1, C_2, D) \) with \( D \in \{\text{near, medium, far}\} \) between cities.

Classification Experiments As a classification setting, we consider the problem of identifying so-called meta-alliances in Travian, introduced by Karwath et al. [9]. A meta-alliance is a group of alliances that closely cooperate. We manually identified meta-alliances in the collected game data based on the alliance names for instance the alliances 'A-', '+A-', and '-A-' are different wings of the same meta-alliance.

From all available game data 30 sequences were extracted constituting positive examples. A further 60 negative examples were obtained by giving the wrong meta-alliance information.

We hand-coded a simple CPT-theory that encodes a few basic features that player of the same meta-alliances behave like alliance partners, for example

\[
\text{conq}(C, P_1) : 0.0061 \lor \text{nil} : 0.9939 \leftarrow \text{city}(C, \prec P_2), \text{player}(P_2, \prec A_1), \text{player}(P_1, \prec A_2), \neg \text{alliance partners}(A_1, A_2).
\]

states that the player \( P_1 \) attacks a city \( C \) of a player \( P_2 \) who is not his alliance partner.
A CPT-theory can be used for classification given positive and negative training sequences by learning the parameters of two CPT-theories $T_+$ and $T_-$. The likelihood of a test sequence $S$ under the positive and negative models, $P(S \mid T_+)$ and $P(S \mid T_-)$ is evaluated, and the class for which this likelihood is higher is predicted.

Figure 4 compares the results obtained for CPT-L with those of the BOOSTEDREAL system [9]. BOOSTEDREAL is a state-of-the-art system for classification of (relational) sequences by alignment, which uses a discriminative approach based on boosting the reward model used in the alignment algorithm [9]. Even so BOOSTEDREAL is a discriminative and much more complicated approach tailored toward classification problems, it does not perform significantly better. Furthermore are the resulting models harder to interpret, as the boosted reward function is represented as an ensemble of relational regression trees. Figure 4 shows that CPT-L, at 82.22% with standard deviation of 9.37, achieves a slightly but not significantly lower accuracy than the best observed result for BOOSTEDREAL. Overall, we can conclude from this experiment that even with the simple rule set used, CPT-L is able to learn a model that captures useful information about the positive and negative class, and achieves similar accuracies as other state-of-the-art sequence classification schemes. Learning a single model in this domain takes under 2 minutes.

Prediction Experiments We now consider the problem of predicting player actions within Travian. From all available data, we again extracted 30 sequences involving 10 players and 30–40 cities, which are tracked over a period of one month. Player sets are
chosen such that there are no interactions between players in different sets, but a high number of interactions between players within one set.

We defined a world model in CPT-L that expresses the probability for player actions such as conquests of cities and changes in alliances affiliation, and updates the world state accordingly. Player actions in Travian—although strongly stochastic—are typically explainable from the social context of the game: different players from the same alliance jointly attack a certain territory on the map, there are retaliation attacks at the alliance level, or players leave alliances that have lost many cities in a short period of time. From a causal perspective, actions are thus triggered by certain (relational) patterns that hold in the game graph. Such patterns can be naturally expressed as bodies of rules which trigger actions encoded in the head of the rule. As an example, consider the rules

\[
\text{conq}(P, C) : 0.039 \lor \text{nil} : 0.961 \quad \text{conq}(P, C'), \text{city}(C', P') \quad \text{city}(C, P') \quad \text{conq}(P', C) : 0.011 \lor \text{nil} : 0.989 \quad \text{city}(C, P'), \text{allied}(P, A), \text{allied}(P', A), \text{conq}(P', C'), \text{city}(C', P'')
\]

The first rule encodes that a player is likely to conquer a city of a player he or she already attacked in the previous time step. The second rule generalizes this pattern: a player \( P \) is likely to attack a city \( C \) of player \( P' \) if an allied player has attacked \( P' \) in the previous time step.

Moreover, the world state needs to be updated given the players' actions. After a conquest attack \( \text{conq}(P, C) \), the city \( C \) changes ownership to player \( P \) in the next time step. Similarly, an \( \text{alliance}_\text{change}(P, A) \) event changes the alliance affiliation of player \( P \) to alliance \( A \) in the next time step.

We now consider predicting the “conquest” action \( \text{conq}(P, C) \) based on a learned model of world dynamics. The collected sequences of game states were split into one training set (December 2007) and two test sets (January and February 2008). Maximum-likelihood parameters of a hand-crafted CPT-theory \( T \) as described above were learned on the training set using EM. Afterwards, the learned model was used to predict the player action \( \text{conq}(P, C) \) on the test data. For every \( t_0 \), every player \( p \) and city \( c \) occurring in \( S \), the probability of the conquest \( \text{conq}(p, c) \) is computed. This probability is obtained using a sampling-based algorithm. The prediction is compared to the known ground truth (whether the conquest event occurred at that time in the game or not). Instead of predicting whether the player action will be taken in the next step, we can also predict whether it will be taken within the next \( k \) step.

Figure 5, left, shows ROC curves for this experiment with different values \( k \in \{1, 2, 3, 4, 5\} \), evaluated on the first test set. Figure 5, right, shows the corresponding AUC values as a function of \( k \) for both test sets. The achieved area under the ROC curve is substantially above 0.5 (random performance), indicating that the learned CPT-theory \( T \) obtains a reasonable ranking of player/city pairs \( (p/c) \). Moreover, the model is able to predict conquest actions several steps in the future.

Predictions for January are slightly more accurate than for February. This is not surprising as the model has been trained from sequences of December 2007, and indicates a slight change in game dynamics over time. In summary, we conclude that player actions in Travian are indeed to some degree predictable from the social context of the game.
and CPT-L is able to learn such patterns from the data. The computational complexity of learning in this task will be analyzed in detail in the next section.

**Scaling Experiments** The scaling behavior of the learning algorithm was analyzed using the learning setting discussed in the last section, and vary the number of players and cities that are present in any given game state. We used 30 sequences each containing up to 50 players, which together controlled up to 269 cities. To evaluate computational complexity, a model was trained on all sequences, using the same rule set as used for the prediction task.

To illustrate the complexity of the resulting problem, one can approximate the size of the ground network that would have been obtained had we grounded the model to a Bayesian or Markov Network as it is typically done for SRL approaches such as CP-logic or Markov Logic Networks. In such a network, nodes correspond to all groundings of predicates using available domain constants. For the largest domain we have considered (involving 50 players and 269 cities), the size of the ground network is approximately 800,000 nodes, indicating that exact inference and learning in this network would be computationally expensive.

Figure 6 shows the time needed to perform inference in CPT-L in the outlined domain as a function of the size of the (hypothetical) ground network, for up to 20 players. This experiments shows that learning is still possible with large networks. This is in contrast to other approaches that try to overcome the excessive size of ground networks by performing approximate inference, as, for example, in Markov Logic Networks [3].

5 Related Work

There are relatively few existing approaches that can probabilistically model sequences of relational state descriptions. CPT-L can be positioned with respect to them as follows.
Fig. 6: Time for performing inference (in the Expectation Step of the EM algorithm) for the Travan prediction task as a function of the domain size. The y-axis shows runtime in seconds. The x-axis shows the number of nodes in the Bayesian network that would result from the grounding of the CPT-theory in this domain.

First, statistical relational learning systems such as Markov Logic [3], CP-logic [4], Probabilistic Relational Models [10] or Bayesian Logic Programs [5] can be used in this setting by adding an extra time argument to predicates (then called *fluents*). However, inference and learning in these systems is computationally expensive: they support very general models including hidden states, and are not optimized for sequential data.

Another approach designed to model sequences of relational state descriptions are relational simple-transition models [11]. A related approach is that by [12], who employs dynamic Markov Logic to represent stochastic relational processes. Inference is carried out in a ground dynamic Bayesian network constructed from the MLN. In contrast to CPT-L, these two approaches focus on domains where the process generating the data is hidden, and inferring these hidden states from observations. This is a significantly harder setting than the fully observable setting discussed in this paper, and therefore typically only approximate inference is possible [11]. However, we feel that also the easier problem where everything is observable is worthy of investigation in its own right. A better understanding of this problem should also provide new insights into the more complex one. In this context, we can mention that an extension of CPT-L to deal with hidden variables is currently under study, where inference is based on a Monte Carlo method, cf. [13].
6 Conclusions and Future Work

We have presented CPT-L, a probabilistic model for sequences of relational state descriptions. In contrast to other approaches that could be used to model such sequences, CPT-L focuses on computational efficiency rather than expressivity. This was illustrated in various experiments in two real-world domains.

There are two main directions for future work. The current exposition assumes that the rules describing possible world dynamics are pre-specified by the user, and only parameters of the resulting transition distributions are estimated from data. However, specifying appropriate rule sets can be non-trivial. Ultimately, we would like to be able to learn both the structure (i.e., rules) and the parameters of the probabilistic transition model from data. We are currently trying to infer rules for CPT-L, using standard rule learners such as Progol or Tertius. Experiments in this direction are promising but preliminary. Finally, we are interested in applying the presented techniques in other challenging application domains.

References