Mining chains of relations ⋆

Foto Aftrati[1], Gautam Das[2], Aristides Gionis[3], Heikki Mannila[4], Taneli Mielikäinen[5], and Panayiotis Tsaparas[6]

[1] National Technical University of Athens,
afrati@softlab.ece.ntua.gr
[2] University of Texas at Arlington,
gdas@cse.uta.edu
[3] Yahoo! Research, Barcelona,
gionis@yahoo-inc.com
[4] University of Helsinki,
mannila@cs.helsinki.fi
[5] Nokia Research, Palo Alto,
taneli.mielikainen@nokia.com
[6] Microsoft Research, Mountain View,
panats@microsoft.com

Abstract. Traditional data mining methods consider the problem of mining a single relation that relates two different attributes. For example, in a scientific bibliography database, authors are related to papers, and we may be interested in discovering association rules between authors based on the papers that they have co-authored. However, in real life it is often the case that we have multiple attributes related through *chains* of relations. For example, authors write papers, and papers belong to one or more topics, defining a three-level chain of relations.
In this paper we consider the problem of mining such relational chains. We formulate a generic problem of finding selector sets (subsets of objects from one of the attributes) such that the projected dataset—the part of the dataset determined by the selector set—satisfies a specific property. The motivation for our approach is that a given property might not hold on the whole dataset, but holds when projecting the data on a subset of objects. We show that many existing and new data mining problems can be formulated in the framework. We discuss various algorithms and identify the conditions when apriori technique can be used. We experimentally demonstrate the effectiveness and efficiency of our methods.

1 Introduction

Analysis of transactional datasets has been the focus of many data mining algorithms. Even though the model of transactional data is simple, it is powerful enough to express many datasets of interest: customers buying products, documents containing words, students registering for courses, authors writing papers, genes expressed in tissues, and many more. A large amount of work has been

⋆ A preliminary version of the paper appeared in ICDM'05.

done on trying to analyze such two-attribute datasets and to extract useful information such as similarities, dependencies, clusters, frequent sets, association rules, etc [2, 5]. At the same time, there have been many attempts to generalize existing data mining problems on datasets with more complex schemas. For instance, multi-relational data mining [15, 17–19, 21] has been considered an extension to the simple transactional data model. However, addressing the problem in the full generality has been proved to be a daunting task.

In this paper, we focus on the specific problem of finding selector sets from one of the attributes of a multi-relational dataset, such that the projections they define on the dataset satisfy a specific property. As an example, consider a dataset with attributes A (authors), P (papers), and T (topics), and relations $R_1(A, P)$ on authors writing papers, and $R_2(P, T)$ on papers concerning topics. An interesting pattern, e.g., "authors x and y frequently write papers together" might not be true for the whole dataset, but it might hold for a specific topic t. Therefore, it is meaningful to consider projections of the bibliographic data on particular topics and search for interesting patterns (e.g., frequent author sets) that occur on the papers of those topics. Additionally, the schema resulting from combining the two relations $R_1(A, P)$ and $R_2(P, T)$ is rich enough so that one can express patterns that go beyond frequent sets and associations. For example, one of the problems we introduce in a later section asks for finding subsets of topics and corresponding authors who have written more papers than anyone else one those topics. Arguably such prolific authors are candidates of being the most *authoritative* researchers on the corresponding topics. Searching for combinations of {topics, authoritative authors} is a new and interesting data mining problem.

In our approach we model datasets as graphs, and patterns to be mined as graph properties. We formulate a generic problem, which in our graph terminology is as follows: *find subsets of nodes so that the subgraph resulting from projecting the data graph on those nodes satisfies a given property.* Our motivation is that the above formulation is a generalization of existing data mining problems, in the sense that commonly studied problems are instances of our generic problem for certain graph properties. Furthermore, in this paper we introduce a number of additional properties—instantiations to our generic problem—that lead to new and challenging problems.

Our contributions can be summarized as follows:

- We introduce a novel approach to mining multi-relational data. Our formulation is quite powerful and it can express many existing problems in data mining and machine learning. For example, finding frequent itemsets, association rules, as well as classification problems can be cast as special cases of our framework.
- In addition to expressing already existing problems, the proposed framework allows us to define many new interesting problems. We express such mining problems in terms of graph properties. We discuss many examples of specific problems that can be used to obtain useful results in real applications and datasets.

- We give conditions under which monotonicity properties hold, and thus, a level-wise method like apriori (see, e.g., [47]) can be used to speed-up the computations. Many of the problems we consider are NP-hard — many of them are hard instances of node removal problems [60]. For such problems we propose an Integer Programming (IP) formulation that can be used to solve medium-size instances by using existing IP solvers.
- To demonstrate the utility of our model we perform experiments on two datasets: a bibliographic dataset, and the IMDB dataset. Our experiments indicate that our algorithms can handle realistic datasets, and they produce interesting results.

The general problem we consider can be defined for complex database schemas. However, for concreteness we restrict our exposition in cases of three attributes connected by a chain of two relations—as in the example of the bibliographic dataset. Such an extension is one of the simplest that one can make to the traditional transactional model. However, even this restricted setting can be useful in modeling many interesting datasets, and the resulting problems are computationally hard. Thus, we believe that exploring the simple model of two-relation chains can provide valuable insights before proceeding to address the problem for more complex multi-relational schemas. In this paper, we only discuss briefly extensions to more complex schemas in Section 3.4.

The rest of the paper is organized as follows. We start our discussion by presenting the related work in Section 2. In Section 3 we formally define our data mining framework and we give examples of interesting problems. In Section 4.1 we demonstrate a characterization of monotonicity that allows us to identify when a problem can be solved efficiently using a level-wise pruning algorithm. In Section 4.2 we describe Integer Programming formulations that allows us to solve small- and medium-size instances for many of our problems. Section 4.3 contains more details about the algorithms we implement and in Section 5 we discuss the results of our experiments. Finally Section 6 is a short conclusion.

2 Related work

Mining of frequent itemsets and association rules on single binary tables such as market basket databases has been a very popular area of study for over a decade [2, 5]. There has also been some effort on investigating data mining problems at the other end of the spectrum, i.e., *multi-relational mining* [15, 17–19, 21, 11, 24, 32, 10, 34, 8]. The approach taken by researchers has been to generalize apriori-like data mining algorithms to the multi-relational case using inductive logic programming concepts. Our work also has connections with work in mining from multidimensional data such as OLAP databases [53] and with the more recent multi-structural databases [22]. In the latter, algorithms are presented for very general analytical operations that attempt to select and segment the data in interesting ways along certain dimensions. While such approaches have been extremely interesting in concept, our goal is more focused—we proceed from

a single table case to the special case of multiple tables defined by chains of relations which often occur in real-world problems.

The work closest to our work is [35] where the authors introduce *compositional data mining* where they cascade data mining primitive operations over chains of relations. The primitive operations they consider is a bi-clustering operation and a re-description operation. Informally they look for patterns (bi-clusters) that emerge in one relation, after applying operations up in the chain of relations. The re-description operator is similar to to the selection predicates we consider, making their work closely related to ours. However, their work does not aim to optimize the selection process as in our case, but rather enumerate all possible mined patterns.

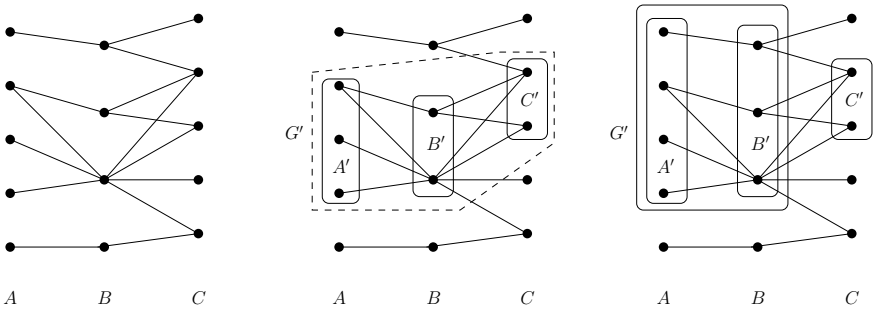
Our work on mining layered graphs also has connections with the widely studied general area of *graph mining*. Various types of graph mining problems have been investigated, such as mining frequent subgraphs [16, 29, 28, 30, 31, 33, 41, 56, 57, 59, 58, 61, 36, 63], link analysis of web graphs [50, 39, 62, 51, 6, 49, 20], extraction of communities [25, 4, 55, 40, 44], identification of influencers [37, 38, 1, 42, 13, 46, 43, 12, 54, 26], and so on. The work in [45] tries to summarize k-partite graphs, by defining clusters per level. As with multi-relational mining, our approach is more focused than these general efforts—we specifically investigate layered graphs, making the case that many interesting real-world problems can be modeled using such graphs, and develop interesting algorithmic techniques that can leverage the structure of such graphs. We also approach the problem from a different perspective, since we focus on the problem of finding selectors that make patterns emerge in the projected datasets, rather than looking for patterns in the whole dataset.

3 The general framework

In the most general case of a database schema we assume attributes $A_1, A_2, \ldots, A_n$ and relations $R_1, R_2, \ldots, R_m$ on the attributes. Transactional data, the object of study of most data mining algorithms, can be viewed as an elementary schema having two attributes A (items) and B (transactions) and a single binary relation $R(A, B)$ (transactions contain items). There are at least three different, but equivalent, ways to view the relation R: (i) a usual database table T on A and B, (ii) a binary co-occurrence matrix M, with rows on A and columns on B, such that $M[a, b]$ is 1 if $(a, b) \in R$ and 0 otherwise, and (iii) a bipartite graph $G = (A, B; E)$ with edges $(a, b) \in E$ if and only if $(a, b) \in R$. In this paper, we find it convenient to work with the graph representation of schemas.[7]

As we noted in the introduction, we focus on a simple extension of the model to three attributes A, B and C and a chain of two relations $R_1(A, B)$ and $R_2(B, C)$. Thus, we assume a graph $G = (A, B, C; E_1, E_2)$ with three sets of nodes A, B and C corresponding to the three attributes and having one node for each value in the domain of the attribute. The graph also has two sets of

[7] Graph representation works well as long as all relations have two attributes. For relations with more than two attributes, one would need to talk about hypergraphs.



(a) The 3-level representation (b) The conjunctive case (c) The disjunctive case

Fig. 1. A graphical representation of the general framework: selecting a subset of nodes in the third level induces a bipartite subgraph between the first two levels. In this example, the *conjunctive interpretation* has been used to define the induced subgraph.

edges, E_1 connecting nodes in A and B, and E_2 connecting nodes in B and C. We call such a graph a *three-level graph*.

Examples of datasets that can be modeled with a three-level graph structure include: AUTHORS writing PAPERS about TOPICS; Web USERS answering online QUESTIONS associated with TAGS; ACTORS playing in MOVIES belonging to GENRES; TRANSCRIPTION-FACTOR-BINDING-SITES occurring at the promoter sequences of GENES that are expressed in TISSUES; and DOCUMENTS containing PARAGRAPHS containing WORDS.

The general data mining framework we consider is graphically depicted in Figure 1, and it is informally defined as follows. Consider the three-level graph $G = (A, B, C; E_1, E_2)$ shown in Figure 1(a). Given a subset $C' \subseteq C$ of nodes from level C, one can induce a subgraph G' from G by taking $B' \subseteq B$ and $A' \subseteq A$, such that every node in A' and B' is reachable from a node in C'. There are (at least) two different ways to define the sets A' and B' depending on whether we require that every node in A' and B' is reachable by *every* node in C' (the conjunctive case – Figure 1(b)), or that every node in A' and B' is reachable by *some* node in C' (the disjunctive case – Figure 1(c)). In each case, we obtain a different subgraph G', with different semantics. Now we are interested on whether the induced subgraph G' satisfies a given property, for example, "G' contains a clique $K_{s,t}$", or "all nodes in A' have degree at least k". The intuition is that the induced subgraph corresponds to a projection of the data, while the graph property corresponds to an interesting pattern. Thus, the generic data mining problem is the following: given a specific property Ψ, to find the selector set C' so that the induced subgraph G' satisfies Ψ.

3.1 Motivation

In this section we discuss the motivation behind our definition and we provide evidence that common data mining and machine learning problems can be cast in our framework.

First consider the traditional transactional data model, e.g., market-basket data. In the graph representation, the data form a bipartite graph $G = (I, T; E)$ with items in the set I, transactions in the set T, and an edge $(i, t) \in E$ if transaction $t \in T$ contains the item $i \in I$. Consider now the problem of finding a frequent itemset of s items with support threshold f. Such an itemset should appear in at least $f|T|$ transactions, giving rise to a $K_{s,f|T|}$ bipartite clique in the graph G. Thus, the problem of finding frequent itemsets corresponds to finding cliques in the bipartite data graph. Furthermore, answering the question whether the dataset contains a frequent itemset of size s with support f, corresponds to answering whether the input graph G contains a $K_{s,f|T|}$ clique. In other words, it corresponds to testing a *property* of the graph G.

Another well-studied problem in data mining is the problem of finding association-rules. An association rule $A \Rightarrow B$ with confidence c holds in the data if the itemset B appears in at least a c-fraction of the transactions in which the itemset A appears. Assume now that we want to find association rules with $|A| = k$ and $|B| = s$. We will show how this problem can be formulated as a selection problem in the three-level framework. Consider the graph representation $G = (I, T; E)$ of the transaction data, as defined before. Now, consider a set of items $A \subseteq I$. The set A induces a subgraph $G_A = (I_A, T_A; E_A)$, with $T_A = \{t : (i, t) \in E \text{ for all } i \in A\}$, $I_A = \{j : j \notin A, \text{ and } (j, t) \in E \text{ for some } t \in T_A\}$, and $E_A = \{(i, t) \in E : i \in I_A \text{ and } t \in T_A\}$. In other words, the subgraph G_A induced by A contains all transactions (T_A) that contain all items in A and all other items (I_A) in those transaction except those in A. The task is to find itemsets A (of size k) such that the induced subgraph G_A contains a $K_{s,c|T_A|}$ clique. The itemset B on the item side of the clique, together with the itemset A define an association rule $A \Rightarrow B$, with confidence c. So, the problem of finding association rules can be formulated as selecting a set of nodes so that the induced subgraph satisfies a given property.

In our next example, we show how a canonical machine-learning problem can also be formulated in our framework, and this time as a three-level graph problem. Consider a dataset of n "examples" $\mathcal{E} = \{\langle \mathbf{d}_i : c_i \rangle, i = 1, \ldots, n\}$, where each example is defined by a datapoint $\mathbf{d}_i$ over a set of attributes and c_i is a class label from a small set C of labels. Think of $\mathbf{d}_i$ as a person's entries to a credit card application questionnaire and the label c_i recording if the credit card was granted or not. The learning problem is to find a set of rules that correctly predict the credit card granting decision for a new applicant $\mathbf{x}$. For instance, such a rule combination could be "**if** `x.income > 50K` **and** `x.age` $\geq$ `18` **then** `yes`".

We now map the above learning problem to a three-level graph mining problem. The graph $G = (C, D, R; E_1, E_2)$ is constructed as follows. The examples in $\mathcal{E}$ induce the subgraph $(C, D; E_1)$. The set C consists of all possible class labels.

Alternatively the class labels can be considered as the possible properties of the data points. D is the set of datapoints, i.e., D has one vertex for each datapoint $\mathbf{d}_i$. There is an edge $(c, d) \in E_1$ iff the datapoint $d \in D$ has property $c \in C$.

The vertex set R captures the set of potential rules (or features). A rule r is a mapping $r : D \to \{0, 1\}$. For example if r is the rule "$\texttt{x.age} \geq \texttt{18}$", then $r(d) = 1$ for all datapoints that correspond to applicants older than 18, and $r(d) = 0$ for all applicants under 18. If the rules are restricted to be in a specific class, say, conjunctions of conditionals on single attributes of size at most three, then one can enumerate all potential rules. There is an edge $(d, r) \in E_2$ iff $r(d) = 1$. Hence, in the disjunctive interpretation, a subset of R induces a disjunction of rules, while in the conjunctive interpretation a conjunction of rules.

There are many classifier learning tasks that can be formulated for such three-level graph by posing additional constraints on the vertex and edge sets. Let us consider the credit card example mentioned above. For simplicity, we assume that there are only two classes, $C = \{\texttt{yes}, \texttt{no}\}$ corresponding on whether the credit card was granted. A person $d \in D$ is connected to the class `yes` if the person's credit card was approved and `no` if the card was declined. Hence, a person can be connected to one, two or none of the classes. There are a few natural formulations of the learning task. For example, the goal in the learning can be to find the set of rules that captures all persons connected to the class `yes` and no persons connected the class `no`, i.e., to find the *consistent classifier* characterizing the people who have been granted the credit card. Note that necessary condition of such set to exist is that each person is connected exactly to one of the classes. In practice there are often misclassifications and multiple class labels for the same data point. Hence, a more practical variant of the learning task would be to construct a rule set that captures people who should (should not) be granted the credit card, i.e., the people who are connected only to the class `yes` (`no`).

The classification problem example is only meant to convey intuition and motivation by casting a well known problem in our general framework. Many important issues such as selecting the collection of rules, avoiding overfitting the data, etc., are not discussed here. However a further discussion of this problem and precise definitions can be found in subsequent sections (Section 3.3 and Section 4.3).

3.2 Problem definition

Before proceeding to formally define our problem, we make a comment on the notation: as a working example in the rest of the paper we use the bibliography dataset (AUTHORS – PAPERS – TOPICS). Therefore, we appropriately denote the three attributes appearing in the formulation by A, P, and T. The names of the problems and the graph properties are also inspired by the bibliography dataset, but this is only for improving the readability—most of problems are meaningful to many other datasets.

We start with attributes A, P and T, relations $E_1(A, P) \subseteq A \times P$ and $E_2(P, T) \subseteq P \times T$, and the corresponding three-level graph $G = (A, P, T; E_1, E_2)$.

Let $S \subseteq T$ be a subset of T. The set S acts as a selector over the sets P and A. First, for a single node $t \in T$, we define

$$P_t = \{p \in P : (p,t) \in E_2\}, \text{ and}$$
$$A_t = \bigcup_{p \in P_t} A_p = \bigcup_{p \in P_t} \{a \in A : (a,p) \in E_1\}.$$

That is, the sets P_t and A_t are the subsets of nodes in P and A, respectively, that are reachable from the node $t \in T$. (The set A_p is the subset of nodes in A that are reachable from the node $p \in P$.) We can extend the definition to the subsets P_S and A_S that is reachable from the set $S \subseteq T$. Extending the definition to sets requires to define the *interpretation* of the selector S. We consider the following two simple cases.

Disjunctive Interpretation In the disjunctive interpretation ($\mathcal{D}$), the subsets P_S and A_S are the set of nodes that are reachable from *at least* one node in S. Therefore, we have

$$P_S^{\mathcal{D}} = \bigcup_{t \in S} P_t \quad \text{and} \quad A_S^{\mathcal{D}} = \bigcup_{p \in P_S^{\mathcal{D}}} A_p.$$

Conjunctive Interpretation In the conjunctive interpretation ($\mathcal{C}$), the subsets P_S and A_S are the set of nodes that are reachable from *every* node in S. Therefore, we have

$$P_S^{\mathcal{C}} = \bigcap_{t \in S} P_t \quad \text{and} \quad A_S^{\mathcal{C}} = \bigcup_{p \in P_S^{\mathcal{C}}} A_p.$$

Now, let $\mathcal{I}$ denote the interpretation, which can be either conjunctive ($\mathcal{C}$), or disjunctive ($\mathcal{D}$), or any other possible interpretation. Given the selector set $S \subseteq T$ and the subsets $A_S^{\mathcal{I}}$ and $P_S^{\mathcal{I}}$, we can define the *induced* three-level graph $G_S^{\mathcal{I}} = (A_S^{\mathcal{I}}, P_S^{\mathcal{I}}, S; E_{1,S}^{\mathcal{I}}, E_{2,S}^{\mathcal{I}})$, where

$$E_{1,S}^{\mathcal{I}} = \{(a,p) \in E_1 : a \in A_S^{\mathcal{I}} \text{ and } p \in P_S^{\mathcal{I}}\}, \text{ and}$$
$$E_{2,S}^{\mathcal{I}} = \{(p,t) \in E_2 : p \in P_S^{\mathcal{I}} \text{ and } t \in S\}.$$

We also define the *induced* bipartite subgraph $B_S^{\mathcal{I}} = (A_S^{\mathcal{I}}, P_S^{\mathcal{I}}; E_{1,S}^{\mathcal{I}})$, which consists of the first two levels of G_S.

Hence, the selector set S selects a subset $P_S^{\mathcal{I}}$ of P and the set $P_S^{\mathcal{I}}$ induces the bipartite graph $B_S^{\mathcal{I}}$ by selecting all edges in E_1 and nodes in A that are adjacent to some node in $P_S^{\mathcal{I}}$, regardless of the interpretation. (There is no need for any additional interpretations for $A_S^{\mathcal{I}}$ or $B_S^{\mathcal{I}}$ as any further restrictions for $B_S^{\mathcal{I}}$ can be implemented as additional requirements to the property Ψ that $B_S^{\mathcal{I}}$ is required to satisfy.)

We are interested in finding selector sets S for which the induced subgraph $G_S^{\mathcal{I}}$ satisfies certain properties. Let $\mathcal{L}_G^{\mathcal{I}} = \{G_S^{\mathcal{I}} : S \subseteq T\}$ denote the set of all possible induced three-level graphs under interpretation $\mathcal{I}$. We define a property

Ψ as any subset of the set $\mathcal{L}_G^{\mathcal{I}}$. We say that the graph G_S satisfies Ψ if $G_S \in \Psi$. For the following, to ease the notation, we will often omit the superscript $\mathcal{I}$, when it is immaterial to the discussion.

For some specific property Ψ we can define the following data mining problem.

Definition 1 (Ψ Problem). *Given a three-level graph $G = (A, P, T; E_1, E_2)$, and the interpretation $\mathcal{I}$ find a selector set $S \subseteq T$ such that the induced subgraph $G_S^{\mathcal{I}}$ satisfies the property Ψ.*

The definition of the Ψ problem, requires finding any *feasible* solution $S \subseteq T$, such that the graph $G_S^{\mathcal{I}}$ satisfies the property Ψ. It is often the case that there are multiple feasible solutions to the Ψ problem, and we are interested in finding a feasible solution that satisfies an additional requirement, e.g., find the minimal, or maximal selector set $S \subseteq T$ that is a feasible solution to the Ψ problem. Formally, let $g : \mathcal{L}_G \to \mathbb{R}$, be a real-valued function on the set of graphs $\mathcal{L}_G$. We are then interested in finding a feasible solution S, such that the function $g(G_S)$ is optimized. Therefore, we define the following problem.

Definition 2 (g-Ψ Problem). *Given a three-level graph $G = (A, P, T; E_1, E_2)$, and the interpretation $\mathcal{I}$ find a selector set S such that the induced subgraph $G_S^{\mathcal{I}}$ satisfies the property Ψ, and the function g is optimized.*

This problem definition is general enough to capture different optimization problems. For example finding the maximum (or minimum) selector set such that G_S satisfies the property Ψ, corresponds to the case where $g(G_S) = |S|$, and we want to maximize (or minimize $g(G_S)$).

3.3 Examples of properties

In this section we provide examples of interesting properties, some of which we will consider in the remainder of the paper. For the following definitions, we assume that the graph $G = (A, P, T; E_1, E_2)$ is considered as input. Additionally most of the properties require additional input parameters, i.e., they are defined with respect to threshold parameters, prespecified nodes of the graph, etc. Such parameters are mentioned explicitly in the definition of each property.

Given a selector set $S \subseteq T$ we have already defined the three-level induced subgraph $G_S = (A_S, P_S, S; E_{1,S}, E_{2,S})$, and the induced bipartite counterpart $B_S = (A_S, P_S; E_{1,S})$ (for some interpretation, whose index we omit here). Several of the properties we define, are actually properties of the bipartite graph B_S.

- AUTHORITY(c): Given a node $c \in A$, the graph $G_S = (A_S, P_S, S; E_{1,S}, E_{2,S}) \in \mathcal{L}_G$ satisfies AUTHORITY(c) if $c \in A_S$, and c has the maximum degree among all nodes in A_S. That is, given a specific author $c \in A$ we want to find a set of topics S for which the author c has written more papers than any other author, and thus, author c qualifies to be an authority for the combination of topics S. In a Questions'n Answers (QNA) system, where users answer questions online, we are interested in finding the set of TAGS for which a certain user c has answered the most questions.

- BestRank(c): Given a node $c \in A$, the graph $G_S = (A_S, P_S, S; E_{1,S}, E_{2,S}) \in \mathcal{L}_G$ satisfies BestRank(c) if $c \in A_S$, and for every other graph $G_R \in \mathcal{L}_G$, c is *ranked* at least as highly in G_S as in G_R. The rank of a node c in a graph G_S is the number of nodes in A_S with degree strictly higher than the degree of c, plus 1. This property is meant to be a relaxation of the Authority(c) property: since for a specific author c there might be no combination of topics on which c is an authority, we are interested in finding the combination T_c of topics for which author c is the "most authoritative". There might be other authors more authoritative than c on T_c but this is the best that c can do.
- Clique: The graph $G_S \in \mathcal{L}_G$ satisfies Clique if the corresponding bipartite graph B_S is a bipartite clique. Here we are interested in topics in which all papers have been written by the same set of authors. This property is more intuitive for the case of a biological dataset consisting of attributes tissues-genes-TFBSs, where we look for TBFS's which regulate genes that are all expressed over the same tissues. It also makes sense in the QnA setting, where we are looking for a set of tags that define communities of users that answer the same questions.
- Frequency(f, s): Given threshold value $f \in [0, 1]$, and an integer value s the graph $G_S = (A_S, P_S, S; E_{1,S}, E_{2,S}) \in \mathcal{L}_G$ satisfies the property Frequency(f, s) if the corresponding bipartite graph B_S contains a bipartite clique $K_{s,f|P_S|}$. The intuition here is that a bipartite clique $K_{s,f|P_S|}$ implies a frequent itemset of size s with frequency threshold f on the induced subgraph. For this property, it is also interesting to consider the g-Ψ problem, where we define the objective function g to be the number of $K_{s,f|P_S|}$ cliques, and then look for the selector set that maximizes the value of the function g, that is, it maximizes the number of frequent itemsets.
- Majority: The graph $G_S = (A_S, P_S, S; E_{1,S}, E_{2,S}) \in \mathcal{L}_G$ satisfies Majority if for every $a \in A_S$ we have $|E^a_{1,S}| \geq |E^a_1 \setminus E^a_{1,S}|$, that is, for every node a in A_S, the majority of edges in G incident on a are included in the graph G_S. In the author-paper-topic context this means that in the induced subgraph for each selected author, the majority of its papers are selected by the selector topic set.
- Popularity(b): Given a positive integer b, the graph $G_S = (A_S, P_S, S; E_{1,S}, E_{2,S}) \in \mathcal{L}_G$ satisfies Popularity(b) if $|A_S| \geq b$. That is, we want to find topics for which more than b authors have written papers about.
- Impact(b): Given a positive integer b, the graph $G_S = (A_S, P_S, S; E_{1,S}, E_{2,S}) \in \mathcal{L}_G$ satisfies Impact(b) if for all nodes $a \in A_S$, the degree of a in the induced subgraph G_S is at least b. Here, the intention is to search for topics on which all authors have written at least b papers—and thus, hopefully, also have impact.
- AbsoluteImpact(b): Given a positive integer b, a graph $G_S = (A_S, P_S, S; E_{1,S}, E_{2,S}) \in \mathcal{L}_G$ satisfies AbsoluteImpact(b) if for all nodes $a \in A_S$, the degree of a in G is at least b. Note that the difference with the previous definition

is that we now consider the degree of node a in the graph G, rather than the induced subgraph G_S.

- CollaborationClique: A graph $G_S = (A_S, P_S, S; E_{1,S}, E_{2,S}) \in \mathcal{L}_G$ satisfies the property CollaborationClique if for every pair of nodes $a, b \in A_S$, there exists at least one node $p \in P_S$, such that $(a, p) \in E_{1,S}$ and $(b, p) \in E_{1,S}$. In other words, each pair of authors have co-authored at least one paper on the topics of S.
- Classification(c): In this setting we assume that the first level A is the set of class labels, the second level P is the set of examples, and the third level T is the set of features. Given a node $c \in A$, a graph $G_S = (A_S, P_S, S; E_{1,S}, E_{2,S}) \in \mathcal{L}_G$ satisfies Classification(c) if $P_S = \{p \in P : (c, p) \in E_1\}$ and $A_S = \{c\}$. That is, the selector set, must be such that an example $p \in P$ is selected if and only if it belongs to class c. Note that this implicitly assumes that each example is associated with a single class label, otherwise there is no feasible solution. Weaker properties can also be defined, if we allow some of the examples of other classes to be selected, or if we do not require all of the examples of class c to be selected. Those weaker versions can be defined using constraints on the number of false positives and false negatives. Also, one can look for feature sets characterizing multiple classes or combinations of classes, hence being related to multi-task learning [9].
- ProgramCommittee(Z, l, m): For this property, we break the convention that the selector operates on the set of topics, and we will assume that we select from the set of authors. This does not change anything in our definitions, since we can just swap the roles of the sets A and T. We are given a set $Z \subseteq T$ (topics of a conference), and values l and m. We say that the induced subgraph $G_S = (S, P_S, T_S; E_{1,S}, E_{2,S}) \in \mathcal{L}_G^{\mathcal{D}}$ satisfies the property ProgramCommittee(Z, l, m) if $T_S = Z$ (exactly the given topic set is selected), $|S| = m$, (m members in the program committee), and every node $t \in Z$ is connected to at least l nodes in S (for each topic there are at least l experts in the committee). Notice that this is the only example where we make use of the selector set S to define the property. Also, this is the only example in which we need to specify the interpretation $\mathcal{I}$, since the problem makes little sense in the case of the conjunctive interpretation.

3.4 Extensions of the model

There are several ways in which our model can be extended to include more complex cases. Here we outline some of the possible extensions.

Boolean interpretations in between of disjunctive and conjunctive interpretations Disjunctive and conjunctive interpretations are two extreme ways of selecting the nodes in P. Let S be the selector set. A node $p \in P$ belongs in $P_S^{\mathcal{D}}$ if and only if there is at least one edge from p to a node in S, and $p \in P$ belongs in $P_S^{\mathcal{D}}$ if and only if $(p, t) \in E$ for each $t \in S$. Hence, $P_S^{\mathcal{D}}$

contains all nodes in P covered by S and $P_S^{\mathcal{C}}$ contains all nodes in P that form a bi-clique with S.

There is a natural interpretation in between of these two extreme interpretations that unifies both the disjunctive and conjunctive interpretations. In the unifying interpretation the set $P_S^\delta \subseteq P$ of elements selected by S consists of all elements in P that are connected to at least δ-fraction of the nodes in S, i.e.,

$$P_S^\delta = \{p \in P : |\{t \in S : (p,t) \in E\}| \geq \delta |S|\}.$$

The conjunctive interpretation is obtained by setting δ to be 1, and the disjunctive interpretation by setting $\delta = 1/|S|$.

Weighted graphs In our definitions so far we have assumed that the graphs (or relations) are not weighted. A natural extension is to consider the case that the edges between the nodes of the various levels have weights, that is, the tuples in the corresponding relations E_1 and E_2 are associated with a weight. These weights carry some semantics, and we should modify our definitions to take them into account.

If there is a weight $w(p,t) \in (0,1]$ for each edge $(p,t) \in E_2$, the selection of node $p \in P$ can be done similarly as in Section 3.4: p is selected iff

$$\sum_{(p,t)\in E_2, t\in S} w(p,t) \geq \delta |S|.$$

Consider the case that each edge $(p,t) \in E_2$ is associated with a probability $Pr(p,t)$, and an element $t \in T$ selects a node p with probability $Pr(p,t)$. In that case we can express probabilistic versions of the interpretations. In the conjunctive interpretation, given a selector set S, we have that $p \in P_S$ with probability

$$\prod_{t:(p,t)\in E_2} Pr(p,t),$$

while in the disjunctive interpretation we have that $p \in P_S$ with probability

$$1 - \prod_{t:(p,t)\in E_2} (1 - Pr(p,t)).$$

We can also assume that relation E_1 is weighted. For example, in a dataset of tissues-genes-TBFS's, we may also have information about the expression levels of each gene on each tissue. There are several problems that generalize nicely in this setting, such as, Authority, BestRank, Majority, Impact, AbsoluteImpact. These problems involve looking at the degree of a node $a \in A$, which can naturally be replaced by the weighted degree, and the rest of the definition carries through. Furthermore, we can associate weights to the nodes of the graph, to model, e.g., the costs or the importance of the nodes.

It is also interesting to consider the Classification problem in the weighted case, where we assume that weight of an edge $(c,d) \in E_1$ is the probability that

the example d belongs to class c. We can redefine the selection problem, to look for a set S of features such that the probability of the examples to belong to the chosen class c in the subgraph induced by S is maximized.

More complex schemas The focus of this paper has been on mining three-level chains of relations. However, our definitions can be naturally extended into more complex schemas, involving more attributes and relations. In this general setting we have m attributes $A_1, \ldots, A_m$, and k binary relations $E_1, \ldots, E_k$. Thus we obtain a graph $G = (A_1, \ldots, A_m; E_1, \ldots, E_k)$. We assume that the relations are such that the resulting graph is connected. If A_s is the selector attribute, a node $s \in A_s$ selects a node in another attribute A_i, if there is a path between them in the graph G. Given a selector set $S \subseteq A_s$, and an interpretation, we can naturally extend the definitions in Section 3.2 to define the induced subgraph G_S, and then look for properties of this graph.

Schemas that would be interesting to explore in future work include the following.

- Longer chains of relations.
- Schemas in the form of k-partite graphs.
- Star schemas, where the selector attribute is one of the spikes of the star.
- Wheel graphs, where the selector attribute is the center of the wheel.

Implicit topics Sometimes the set of topics can be very large but still it can be decided efficiently whether a given paper p is connected to a given topic t, e.g., in polynomial time in the size of the graph $(A, P; E_1)$.

This is the case, for example, in learning boolean formulas in disjunctive (or conjunctive) normal form. Namely, for each topic $i \in T$ there is a monomial m_i over ℓ variables and there is a binary vector $b_j \in \{0,1\}^\ell$ associated to each paper $j \in P$. A topic $i \in T$ is connected to a paper $j \in P$ if and only if b_j satisfies the m_i. Hence, the problem corresponds the Classification problem (see Section 4.3) where the topics and their links to the papers are not given explicitly but by a polynomial-time algorithm determining for any topic $t \in T$ and paper $p \in P$ whether or not $(p, t) \in E_2$.

4 Algorithmic tools

In this section we study characteristics of the various properties, and we show how they can help us in performing data mining tasks efficiently. We identify cases where level-wise methods (like the apriori algorithm) can be used and we propose an integer programming formulation that can be used in many problems. Finally we focus in four specific problems and discuss methods for their solution in more detail.

4.1 A characterization of monotonicity

The objective in this subsection is to identify cases where one can use standard level-wise methods, like the apriori algorithm and its variants. Given a three-level graph $G = (A, P, T; E_1, E_2)$, and an interpretation $\mathcal{I} \in \{\mathcal{C}, \mathcal{D}\}$, recall that $\mathcal{L}^{\mathcal{I}}$ is the set of all possible induced graphs under interpretation $\mathcal{I}$, for all possible selector sets $S \subseteq T$. We first give the following definitions for *monotonicity* and *anti-monotonicity*.

Definition 3. *A property Ψ is* monotone *on the set $\mathcal{L}^{\mathcal{I}}_G$ if the following is true: if for some selector set $S \subseteq T$ we have $G^{\mathcal{I}}_S \in \Psi$, then for all $R \subseteq S$ we have $G^{\mathcal{I}}_R \in \Psi$.*

A property Ψ is anti-monotone *on the set $\mathcal{L}^{\mathcal{I}}_G$ if the following is true: if for some selector set $S \subseteq T$ we have $G^{\mathcal{I}}_S \in \Psi$, then for all $R \supseteq S$ we have $G^{\mathcal{I}}_R \in \Psi$.*

The concept of monotonicity can be used to gain considerable efficiency in the computations by enumerating all possible sets of selectors in an incremental fashion (generate a set after having generated all of its subsets). Once it is found that the property Ψ is not satisfied for some selector set S, then the search space can be pruned by discarding from consideration all supersets of S. Many different implementations of this idea can be found in the literature [5]. Here we relate monotonicity and anti-monotonicity with the concept of *hereditary* properties on graphs.

Definition 4. *A property Ψ is* hereditary *on a class $\mathcal{G}$ of graphs with respect to node deletions, if the following is true: if $G = (V, E)$ is a graph that satisfies Ψ, then any subgraph $G' = (V', E')$ of G, induced by a subset $V' \in V$ also satisfies the property.*

A property Ψ is anti-hereditary *on a class $\mathcal{G}$ of graphs with respect to node deletions, if the following is true: if $G = (V, E)$ is a graph that does not satisfy Ψ, then any subgraph $G' = (V', E')$ of G, induced by a subset $V' \in V$ also does not satisfy the property.*

We can show that if a graph property Ψ is hereditary, it implies that the property is also monotone with respect to the disjunctive interpretation and anti-monotone with respect to the conjunctive interpretation.

Theorem 1. *Any hereditary property is monotone on the set $\mathcal{L}^{\mathcal{D}}_G$, and anti-monotone on the set $\mathcal{L}^{\mathcal{C}}_G$.*

Any anti-hereditary property is anti-monotone on the set $\mathcal{L}^{\mathcal{D}}_G$, and monotone on the set $\mathcal{L}^{\mathcal{C}}_G$.

Proof. Consider a hereditary property Ψ, and also consider any selector sets S and R such that $S \subseteq R \subseteq T$. We have $G^{\mathcal{D}}_S \subseteq G^{\mathcal{D}}_R$ and $G^{\mathcal{C}}_R \subseteq G^{\mathcal{C}}_S$. Since Ψ is hereditary it follows that if $G^{\mathcal{D}}_R \in \Psi$ then $G^{\mathcal{D}}_S \in \Psi$. Similarly, if $G^{\mathcal{C}}_S \in \Psi$ then $G^{\mathcal{C}}_R \in \Psi$. Thus, Ψ is monotone on $\mathcal{L}^{\mathcal{D}}_G$, and anti-monotone on $\mathcal{L}^{\mathcal{C}}_G$.

The rest of the theorem follows from the fact that an anti-hereditary property is a complement of a hereditary property.

The implication of the theorem is that, usually, given a property Ψ, one can check easily if Ψ is (anti-)hereditary or not. If it is (anti-)hereditary, then we know that a level-wise algorithm can be devised for solving the graph mining problem for this property [47]. For example, CLIQUE is hereditary, since removing any nodes from a clique graph we are still left with a clique. Additionally, the following results are immediate.

Proposition 1. *The properties* CLIQUE *and* ABSOLUTEIMPACT *are monotone on* $\mathcal{L}_G^{\mathcal{D}}$ *and anti-monotone on* $\mathcal{L}_G^{\mathcal{C}}$. *The property* POPULARITY *is anti-monotone on* $\mathcal{L}_G^{\mathcal{D}}$ *and monotone on* $\mathcal{L}_G^{\mathcal{C}}$.

On the other hand, by constructing simple counterexamples, one can show that the properties AUTHORITY, BESTRANK, FREQUENCY, MAJORITY, IMPACT, CLASSIFICATION and COLLABORATIONCLIQUE are neither monotone nor anti-monotone on $\mathcal{L}_G^{\mathcal{D}}$ or $\mathcal{L}_G^{\mathcal{C}}$. Thus, level-wise methods do not suffice to solve the corresponding problems.

4.2 Integer Programming formulations

Computing the *maximal*, *minimal*, or *any* selector set is an NP-hard problem for most of the examples given in Section 3.3. In Section 4.1 we showed that if the property under consideration is hereditary, then the task of enumerating all solution sets (therefore also the maximal and the minimal sets) can be done efficiently by a level-wise approach.

In this section we give IP formulations for some of the examples given in Section 3.3. Solvers for IP and LP have been in the core of extensive research in operations research and applied algorithms, and highly optimized methods are available [48]. We found that small- and medium-size instances of the problems we consider can be solved quite efficiently using an off-the-shelf IP solver.[8] Notice also that in the IP formulation we typically ask for one solution (often by imposing an objective function to optimize), as opposed to enumerating all solutions like in the previous section.

Let $G = (A, P, T; E_1, E_2)$ denote the three-level graph that represents the relational chain. For each element $i \in T$, we define a variable $t_i \in \{0, 1\}$, where $t_i = 1$ if the element i is selected and zero otherwise. Furthermore for each element $j \in P$ we define a variable $p_j \in \{0, 1\}$. We need also to add constraints on these variables.

First, we implement the selection of elements in P. In the disjunctive interpretation we require that if an element $i \in T$ is chosen, then the set $P_i^T = \{j \in P : (j, i) \in E_2\}$, consisting of all the papers in P that belong to topic i, is also chosen. This condition is enforced by requiring that

$$p_j \geq t_i \text{ for all } j \in P_i^T.$$

[8] In practice, we solve IPs using the Mixed Integer Programming (MIP) solver `lp_solve` obtained from `http://groups.yahoo.com/group/lp_solve/`.

Furthermore, we require that for each $j \in P$ that is chosen, at least one $i \in T$ is chosen, such that $(j,i) \in E_2$. Let $T_j^P = \{i \in T : (j,i) \in E_2\}$ be the set of topics to which paper j belongs. Hence, we have that

$$\sum_{i \in T_j^P} t_i \geq p_j \text{ for all } j \in P.$$

The constraints guarantee that if the variables $t_i \in [0,1]$ take values in $\{0,1\}$ then the variables $p_j \in [0,1]$ will also take values in $\{0,1\}$. Thus, in the disjunctive interpretation we can relax the constraints $p_j \in \{0,1\}$ to $p_j \in [0,1]$ for all $j \in P$.

In conjunctive interpretation we require that a paper in P can be selected if and only if it is connected to all nodes in the selector set S. This can be expressed by the inequalities

$$\sum_{i \in T_j^P} t_i \geq |T_j^P| p_j \qquad \text{and} \qquad |T_j^P| - \sum_{i \in T_j^P} t_i \geq 1 - p_j.$$

The constraints guarantee that if the variables $p_j \in [0,1]$ take values in $\{0,1\}$ then the variables $t_i \in [0,1]$ will also take values in $\{0,1\}$. Thus, in the conjunctive interpretation we can relax the constraints $t_i \in \{0,1\}$ to $t_i \in [0,1]$ for all $i \in T$.

Finally, for each element $k \in A$, we similarly define a variable $a_k \in \{0,1\}$ and impose the same constraints as for the p_j variables in the disjunctive interpretation. Let $A_j^P = \{k : (k,j) \in E_1\}$ be the set of authors of paper j, and $P_k^A = \{j : (k,j) \in E_1\}$ be the set of papers authored by author k. Then we have

$$a_k \geq p_j \qquad \text{and} \qquad \sum_{j \in P_k^A} p_j \geq a_k$$

for all $k \in A_j^P$, and again the constraints $a_k \in \{0,1\}$ can be relaxed to $a_k \in [0,1]$ for all $k \in A$.[9] We also define variable x_k, that captures the degree of the node $k \in A$ in the subgraph induced by the selected nodes in T, i.e., $x_k = \sum_{j \in P_k^A} p_j$.

We now show how to express some of the properties we discussed in Section 3.3 by imposing restrictions on the different variables.

- AUTHORITY(c): We impose the constraints $x_c \geq x_k$ for all $k \in A - \{c\}$. Note that the potential topics are the topics that author c has at least one paper. That is, we can restrict the search for a good topic set to the subgraph induced by the topics of author c.
- CLIQUE: We impose the constraint that $a_k = \sum_{j \in P} p_j$ for all $k \in A$.
- FREQUENCY(f,s): We define variables z_k for selecting a subset of selected authors and y_j for selecting a subset of selected papers. These variables are used to define the clique. First, we express that only selected authors and

[9] In fact, by allowing some of the variables to be real-valued, we can used Mixed Integer Programming (MIP) instead of IP and improve the performance considerably.

papers can be selected. That is, $z_k \in \{0, a_k\}$ for all $k \in A$, and $y_j \in \{0, p_j\}$ for all $j \in P$. Second, we add constraints requiring that the number of authored in the clique is s and that the number of papers in the clique is at least $f|P_S|$, i.e., $\sum_{k \in A} z_k = s$ and $\sum_{j \in P} y_j \geq f \sum_{j \in P} p_j$. Finally, we require that the variables z_k and y_j define a clique: $\sum_{k \in A, (k,j) \in E_1} z_k = s y_j$ for all $j \in P$.

- Majority: We impose the constraint that $(1 - a_k)|P_k^A| + x_k \geq |P_k^A|/2$ for all $k \in A$.
- Popularity(b): We impose the constraint that $\sum_{k \in A} a_k \geq b$.
- Impact(b): We impose the constraint that $x_k \geq b$ for all $k \in A$.
- AbsoluteImpact(b): We impose the constraint that $|P_k^A| \geq b a_k$ for all $k \in A$.
- CollaborationClique: Let us denote the set of co-authors of author $k \in A$ by $C_k = \{k' \in A : \exists j \in P_k^A \text{ s.t. } (k, j), (k', j) \in E_1\}$. Then we impose the constraints $a_k|A| + c - \sum_{k' \in C_k} a_{k'} \leq |A|$ for all $k \in A$, and $c = \sum_{k \in A} a_k$ where c is a real-valued variable.
- ProgramCommittee(Z, l, m): Let $A_i^T = \{k \in A : \exists j \in P_k^A \text{ s.t. } (j, i) \in E_2\}$. We add the constraints $\sum_{k \in A} a_k \leq m$, and $\sum_{k \in A_i^T} a_k \geq l$ for all $i \in Z$. For this problem, we need also the constraints $a_k \in \{0, 1\}$ for all $k \in A$ since there are no topic set selection involved in the program. Note also that we can neglect the authors outside the set $\bigcup_{i \in Z} A_i^T$.

4.3 Case studies

In this subsection, we focus on four specific problems among those listed in Section 3.3 and we look into detailed aspects of their solution. Two of them are selected to perform experiments with on real datasets. These experiments are reported in the next section.

The Frequency problem Recall that the Frequency problem is as follows. Given the graph $G = (A, P, T, E_1, E_2)$ a value s, and a threshold value f, we want to find a subset of nodes $S \subseteq T$ so that in the induced subgraph $G_S = (A_S, P_S, S; E_{1,S}, E_{2,S})$ there exist frequently occurring "itemsets" $V \subseteq A_S$ of size s. In other words, for itemset V to be frequent according our definition, it needs to be the case that V is frequent on the restriction of the graph imposed by a selector set S. Thus, finding frequent itemsets V with frequency threshold f in the three-level graphs is equivalent to fining *association rules* $S \Rightarrow V$ with confidence threshold f.

One only needs to add the restriction that the premise set S is selected from node set T and the conclusion set V is selected from node set A, but this only prunes the possible search space. There are many algorithms for association-rule mining in the literature [5] and any of them would be applicable in our setting with the above-mentioned modification.

The Authority problem For a single author c, we solve the authority problem using MIP. As the optimization objective function g, we consider maximizing the

number of authors related to the topic set $S \subseteq T$, that is, $g(G_S) = |A_S|$, for $G_S = (A_S, P_S, S; E_{1,S}, E_{2,S})$.

The requirement that the author has the largest number of papers in the induced subgraph can sometimes be too restrictive. One could also, for example, minimize the absolute distance between the highest degree $\max_{k \in A_S} x_k$ of the authors and the degree x_c of the author c, or minimize $\sum_{k \in A_S}(x_k - x_c)$.

The rank alone, however, does not tell everything about the authority of an author. For example, the number of authors and papers in the induced subgraph matter. Thus, it makes sense to search for ranks for all different topic sets.

A set of papers fully determines the set of authors and a set of topics fully determines the set of papers. It is often the case that different sets of topics induce the same set of papers. Thus, we do not have to compute the rankings of the authors for all sets of topics to obtain all different rankings; it suffices to compute the rankings only once for each distinct set of papers that results by a combination of topics. The actual details of how to do this depend on which interpretation we use.

Conjunctive interpretation In the conjunctive interpretation, the subgraph induced by a topic set S contains a paper $j \in P$ if and only if $S \subseteq T_j^P$, that is, S is a subset of the set of topics to which paper j belongs. Thus, we can consider each paper $j \in P$ as a topic set T_j^P. Finding all topic sets that induce a non-empty paper set in the conjunctive interpretation can be easily done using a bottom-up apriori approach. The problem can be cast as a frequent-set mining task in a database consisting the topic sets T_j^P of the papers $j \in P$ with frequency threshold $f = 1/|P|$ (so that a chosen topic set is related to at least one paper). Any frequent set mining algorithms can be used, e.g., see [5]. Furthermore, we can easily impose a minimum frequency constraint for the topic sets, i.e., we can require that a topic set should be contained in at least $f|P|$ sets $T_j^P, j \in P$ for a given frequency threshold $f \in [0, 1]$. In addition to being a natural constraint for the problem, this often decreases considerably the number of topic sets to be ranked.

However, it is sufficient to compute the rankings only once for each distinct set of papers. It can be shown that the smallest such collection of topic sets consists of the topic sets $S \subseteq T$ such that $S = \bigcap_{i \in S, j \in P_i^T} T_j^P$. Intuitively, this means that the set S is closed under the following operation: take the set of papers that are connected to all topics in S. Then for each paper j compute T_j^P, the set of topics to which paper j belongs, and then take the intersection of T_j^P's. This operation essentially computes the nodes in T that are reachable from S when you follow an edge from S to P, and then back to T. The intersection of T_j^P's should give the set S. In frequent set mining such sets are known as the closed sets, and there are many efficient algorithms discovering (frequent) closed sets [5]. The number of closed frequent itemsets can be exponentially smaller than the number of all frequent itemsets, and actually in practice the closed frequent itemsets are often only a fraction of all frequent itemsets.

Disjunctive interpretation In the disjunctive interpretation, the subgraph induced by the topic set S contains a paper $j \in P$ if and only if S hits the paper, i.e., $S \cap T_j^P \neq \emptyset$. Hence, it is sufficient to compute the rankings only for those topic sets S that hit strictly more papers than any of their subsets. By definition, such sets of topics correspond to minimal hypergraph transversals and their subsets in the hypergraph $\left(T, \left\{T_j^P\right\}_{j \in P}\right)$, i.e., the partial minimal hypergraph transversals.

Definition 5. *A hypergraph is a pair $H = (X, \mathcal{F})$ where X is a finite set and $\mathcal{F}$ is a collection of subsets of X. A set $Y \subseteq X$ is a hypergraph transversal in H if and only if $Y \cap Z \neq \emptyset$ for all $Z \in \mathcal{F}$. A hypergraph transversal Y is minimal if and only if no proper subset of it is a hypergraph transversal.*

All partial minimal hypergraph transversals can be generated by a level-wise search because each subset of a partial minimal hypergraph transversal is a partial minimal hypergraph transversal. Furthermore, each partial minimal transversal in the hypergraph $\left(T, \left\{T_j^P\right\}_{j \in P}\right)$ selects a different set of papers than any of its sub- or superset.

Theorem 2. *Let $Z' \subsetneq Z \subsetneq Y$ where Y is a minimal hypergraph transversal. Then $P_Z^{\mathcal{D}} \neq P_{Z'}^{\mathcal{D}}$.*

Proof. Let Y be a minimal hypergraph transversal and assume that $Z' \cap Z$ hits all same sets in the hypergraph as Z for some $Z' \subsetneq Z \subsetneq Y$. Then $Y \setminus (Z \setminus Z')$ hits the same set in the hypergraph as Y, which is in contradiction with the assumption that Y is a minimal hypergraph transversal.

The all minimal hypergraph transversals could be enumerate also by discovering all *free* itemsets in the transaction database representing the complement of the bipartite graph $(P, T; E_2)$ where topics are items and papers transactions. (Free itemsets are itemsets that have strictly higher frequency in the data than any of their strict subsets. Free frequent itemsets can be discovered using the level-wise search [7].) More specifically, the complements of the free itemsets in such data correspond to the minimal transversals in a hypergraph $H = (X, \mathcal{F})$:

$$\bigcup \{Z \in \mathcal{F} : Z \cap Y \neq \emptyset\} = X \setminus \bigcap \{X \setminus Z \in \mathcal{F} : Z \cap Y \neq \emptyset\},$$

i.e., that the union of sets $Z \in \mathcal{F}$ intersecting with the set Y is the complement of the intersection of the sets $X \setminus Z \in \mathcal{F}$ such that Z intersects with Y.

In the disjunctive interpretation of the AUTHORITY problem we impose an additional constraint for the topic sets to make the obtained topic sets more meaningful. Namely, we require that for a topic set to be relevant, there must be at least one author that has written papers about all of the topics. This further prunes the search space and eases the candidate generation in the level-wise solution.

The ProgramCommittee **problem** For the exact solution to the ProgramCommittee problem we use the MIP formulation sketched in Section 4.2. That is, we look for a set of m authors such that for each topic in a given set of topics Z there are at least l selected authors with a paper on this topic. Among such sets of authors, we aim to maximize the number of papers of the authors on the topics in Z. To simplify considerations, we assume, without loss of generality, that the topic set T of the given three-level graph $G = (A, P, T; E_1, E_2)$ is equal to Z and that all authors and papers are connected to the topics.

Although the ProgramCommittee problem can be solved exactly using mixed integer programming techniques, one can also obtain approximate solutions in polynomial time in the size of G. The ProgramCommittee problem can be decomposed into the following subproblems.

First, for any solution to the ProgramCommittee problem we require that for each topic in Z there are at least l selected authors with papers about the topic. This problem is known as the minimum set multicover problem [52]:

Problem 1 (Minimum set multicover). Given a collection $\mathcal{C}$ of subsets of S and a positive integer l, find the collection $\mathcal{C}' \subseteq \mathcal{C}$ of the smallest cardinality such that every element in S is contained in at least l sets in $\mathcal{C}'$.

The problem is NP-hard and polynomial-time inapproximable within a factor $(1 - \epsilon) \log |S|$ for all $\epsilon > 0$, unless NP $\subseteq$ Dtime$(n^{\log \log n})$ [23]. However, it can be approximated in polynomial time within a factor $H_{|S|}$ where $H_{|S|} = 1 + 1/2 + \ldots + 1/|S| \leq 1 + \ln |S|$ [52]. Hence, if there is a program committee of size at most m covering each topic in Z at least l times, we can find such a program committee of size at most $mH_{|Z|}$.

Second, we want to maximize the number of papers (on the given set Z of topics) by the selected committee. This problem is known as the maximum coverage problem [23]:

Problem 2 (Maximum coverage). Given a collection $\mathcal{C}$ of subsets of a finite set S and a positive integer k, find the collection $\mathcal{C}' \subseteq \mathcal{C}$ covering as many elements in S as possible.

The problem NP-hard and polynomial-time inapproximable within the factor $(1 - 1/e) - \epsilon$ for any $\epsilon > 0$, unless NP = P. However, the fraction of covered elements in S by at most k sets in $\mathcal{C}$ can be approximated in polynomial time within a factor $1 - 1/e$ by a greedy algorithm [23]. Hence, we can find a program committee that has at least $1 - 1/e$ times the number of papers as the program committee of the same size with the largest number of papers.

Neither of these solutions is sufficient for our purposes. The minimum set multicover solution ensures that each topic has sufficient number of experts in the program committee, but does not provide any guarantees on the number of papers of the program committee. The maximum coverage solution maximizes the number of papers of the program committee, but does not ensure that each topic has any program committee members.

By combining the approximation algorithms for the minimum set multicover and maximum coverage problems, we can obtain an $(1 + H_{|Z|}, 1 - 1/e)$-

approximation algorithm for the ProgramCommittee problem, i.e., we can derive an algorithm such that the size of the program committee is at most $(1 + H_{|Z|}m)$ and the number of the papers of the program committee is within a factor $1 - 1/e$ of the program committee of size m with the largest number of papers. The algorithm is as follows:

1. Select a set $A' \subseteq A$ of at most $mH_{|Z|}$ authors in such a way that each topic in Z is covered by at least l authors (using the approximation algorithm for the minimum set multicover problem). Stop if such a set does not exist.
2. Select a set $A'' \subseteq A$ of m authors that maximizes the coverage of the papers (using the approximation algorithm for the maximum coverage).
3. Output $A' \cup A''$.

In other words, first we select at most $mH_{|Z|}$ member to the program committee in such a way that each topic of the conference is covered by sufficiently many program committee members and then we select authors that cover large fraction of papers on some of the topics of the conference, regardless of which particular topic they have been publishing of.

Clearly, $|A' \cup A''| \leq (1 + H_{|Z|})m$ and the number of papers covered by the sets in $A' \cup A''$ is within a factor $1 - 1/e$ from the largest number of papers covered by any subset of A of cardinality m.

The algorithm can be improved in practice in several ways. For example, we might not need all sets in A to achieve the factor $1 - 1/e$ approximation of the covering the papers with m authors. We can compute the number h of papers needed to be covered to achieve the approximation factor $1 - 1/e$ by the approximation algorithm for the maximum coverage problem. Let the number of paper covered by A' be h'. Then we need to cover only $h'' = h - h'$ papers more. This can be done by applying the greedy set cover algorithm to the instance that does not contain the papers covered by the authors in A'. The set of authors obtained by this approach is at most as large as $A' \cup A''$. The solution can be improved also by observing that for each covered paper only one author is needed and each topic has to be covered by only l authors. Hence, we can remove one by one the authors from $A' \cup A''$ as far as these constraints are not violated.

The Classification problem The classification problem is equal to learning monomials and clauses of explicit features. These tasks correspond to conjunctive and disjunctive interpretations of the Classification problem, respectively.

Conjunctive interpretation Finding the largest (or any) set $F_{\max} \subseteq T$ corresponding to examples $E \subseteq P$ of a certain class $c \in A$ can be easily obtained by taking all nodes in T that contain all examples of class c, if such a subset exists. (Essentially the same algorithm is well-known also in PAC-learning [3].)

The problem becomes more interesting if we set $g(G_S) = |S|$ and we require the solution S that minimizes g. The problem of obtaining the smallest set $F_{\min} \subseteq T$ capturing all examples of class c and no other examples is known to be NP-hard [3]. The problem can be recast as a minimum set cover problem as

follows. Let $\bar{E}_c \subseteq P$ denote the set of examples of all classes other than c. Also let $F_c \subseteq T$ denote the set of features linking to the examples of the class c. Now consider the bipartite graph $B = (\bar{E}_c, F_c; E)$, where $(p,t) \in E$ if $(p,t) \notin E_2$. For any feasible solution S for the classification problem, the features in S must cover the elements in $\bar{E}_c$ in the bipartite graph B. That is, for each $e \in \bar{E}_c$ there exists $f \in S$, such that $(e,f) \in E$, that is, $(e,f) \notin E_2$. Otherwise, there exists an example $e \in \bar{E}_c$ such that for all for all $f \in S$, $(e,f) \in E_2$, and therefore, e is included in the induced subgraph G_S, thus violating the CLASSIFICATION property. Finding the minimum cover for the elements in $\bar{E}_c$ in the bipartite graph B is an NP-complete problem. However, it can be approximated within a factor $1 + \ln |F_c|$ by the standard greedy procedure that selects each time the feature that covers the most elements [14]. (This algorithm is also well-known in the computational learning theory [27].)

Disjunctive interpretation First note that it is straightforward to find the largest set of features, which induces a subgraph that contains only examples of the target class c. This task can be performed by simply taking all features that disagree with all examples of other classes. Once we have this largest set, then one can find the smallest set, by selecting the minimum subset of sets that covers all examples of the class c. This is again an instance of the set cover problem, and the greedy algorithm [14] can be used to obtain the best approximation factor (logarithmic).

5 Experiments

We now describe our experiments with real data. We used information available on the Web to construct two real datasets with three-level structure. For the datasets we used we found it more interesting to perform experiments with the AUTHORITY problem and the PROGRAMCOMMITTEE problem. Many other possibilities of real datasets with three-level graph structure exist, and depending on the dataset different problems might be of interest.

5.1 Datasets

Bibliography datasets We crawled the ACM digital library website[10] and we extracted information about two publication forums: Journal of ACM (JACM) and ACM Symposium on Theory of Computing (STOC). For each published paper we obtained the list of authors (attribute A), the title (attribute P), and the list of topics (attribute T). For topics we arbitrarily selected to use the *second level* of the "Index Terms" hierarchy of the ACM classification. Examples of topics include "analysis of algorithms and problem complexity", "programming languages", "discrete mathematics", and "numerical analysis". In total, in the JACM dataset we have 2 112 authors, 2 321 papers, and 56 topics. In the STOC dataset we have 1 404 authors, 1 790 papers, and 48 topics.

[10] `http://portal.acm.org/dl`

IMDB dataset We extract the IMDB[11] actors-movies-genres dataset as follows. First we prune movies made for TV and video, TV serials, non-English-speaking movies and movies for which there is no genre. This defines a set of "valid" movies. For each actor we find all the valid movies in which he appears, and we enter an entry in the actor-movie relation if the actor appears in one of the top 5 positions of the credits, thus pruning away secondary roles and extras. This defines the actor-movie relation. For each movie in this relation we find the set of genres it is associated with, obtaining the movies-genres relation. In total, there are 45 342 actors, 71 912 movies and 21 genres.

5.2 Problems

The AUTHORITY **problem** For the AUTHORITY problem, we run the level-wise algorithms described in Section 4.3 on the two bibliography datasets and the IMDB dataset. For compactness, whatever we say about authors, papers, and topics, applies also to actors, movies, and genres, respectively. For each author a and for each combination of topics S that a has written a paper about (under the disjunctive or the conjunctive interpretation), we compute the rank of author a for S. If an author a has written at least one paper on *each* topic of S, and a is ranked first in S, we say that a is an *authority* on S. Given an author a, we define the collection of topic sets $\mathcal{A}(a) = \{S : a \text{ is authority for } S\}$, and $\mathcal{A}^0(a)$ the collection of minimal sets of $\mathcal{A}(a)$, that is, $\mathcal{A}^0(a) = \{S : S \in \mathcal{A}\}$, and there is no $S' \in \mathcal{A}$ such that $S' \subsetneq S\}$. Notice that for authors who are not authorities, the collections $\mathcal{A}(a)$ and $\mathcal{A}^0(a)$ are empty.

A few statistics computed for the STOC dataset are shown in Figure 2. In the first two plots we show the distribution of the number of papers, and the number of topics, per author. One sees that the distribution of the number of papers is very skewed, while the number of topics has a mode at 3. We also look at the collections $\mathcal{A}(a)$ and $\mathcal{A}^0(a)$. If the size of the collection $\mathcal{A}^0(a)$ is large it means that author a has many interests, while if the size of $\mathcal{A}^0(a)$ is small it means that author a is very focused on few topics. Similarly, the average size of sets inside $\mathcal{A}^0(a)$ indicates to what degree an author prefers to work on combination of topics, or on single-topic core areas. In the last two plots of Figure 2 we show the distribution of the size of the collection $\mathcal{A}(a)$ and the scatter plot of the average set size in $\mathcal{A}(a)$ vs. the average set size in $\mathcal{A}^0(a)$.

The author with the most papers in STOC is Wigderson with 36 papers. The values of the size of $\mathcal{A}^0$ and the average set size in $\mathcal{A}^0$ for Wigderson is 37 and 2.8, respectively, indicating that he tends to work in many different combinations of topics. On the other hand, Tarjan who is 4th in the overall ranking with 25 papers, has corresponding values 2 and 1.5. That is, he is very focused on two combinations of topics: "data structures" and ("discrete mathematics", "artificial intelligence"). These indicative results match our intuitions about the authors.

[11] `http://www.imdb.com/`

We observed similar trends when we searched for authorities in the JACM and IMDB datasets, and we omit the results to avoid repetition. As a small example, in the IMDB dataset, we observed that Schwarzenegger is an authority of the combinations ("action", "fantasy") and ("action", "sci-fi") but he is not an authority in any of those single genres.

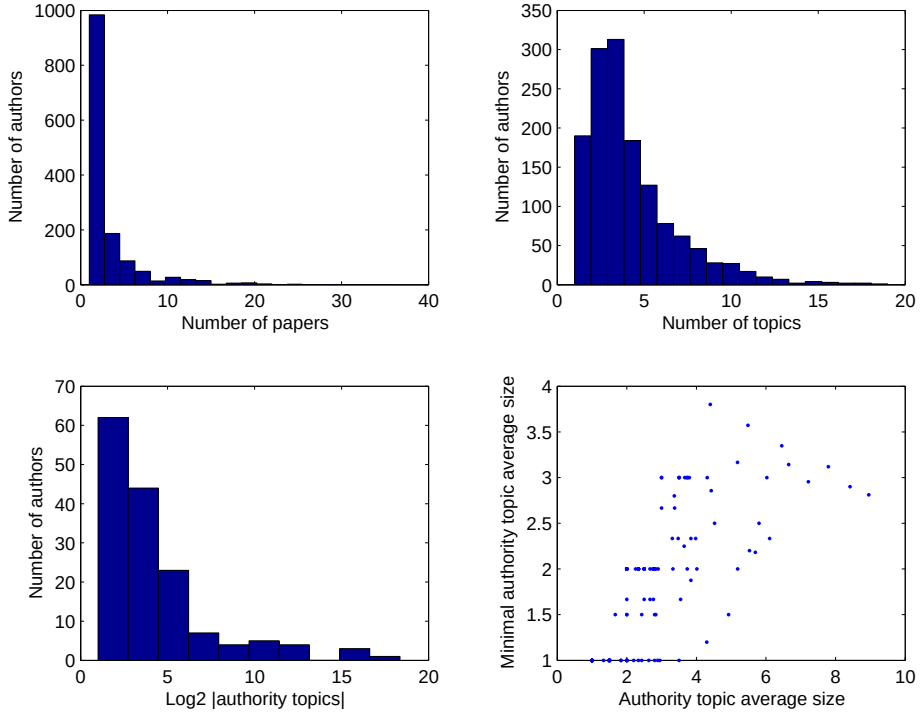


Fig. 2. A few statistics collected on the results from the AUTHORITY problem on the STOC dataset.

The PROGRAMCOMMITTEE problem The task in this experiment is to select program committee members for a subset of topics (potential conference). In our experiment, the only information used is our three-level bibliography dataset; in real life many more considerations are taken into account. Here we give two examples of selecting program committee members for two fictional conferences. For the first conference, which we called LOGIC-AI, we used as seed the topics "mathematical logic and formal languages", "artificial intelligence", "models and principles", and "logics and meanings of programs". For the second conference, which we called ALGORITHMS-COMPLEXITY, we used as seed the topics "discrete mathematics", "analysis of algorithms and problem complexity", "computation

by abstract devices", and "data structures". In both cases we requested a committee of 12 members requiring topics to be covered by at least 4 of the PC members. The objective was to maximize the total number of papers written by the PC members. The committee members for the LOGIC-AI conference, ordered by their number of papers, were

> Vardi, Raz, Vazirani, Blum, Kearns, Kilian,
> Beame, Goldreich, Kushilevitz, Bellare,
> Warmuth, and Smith.

The committee for the ALGORITHMS-COMPLEXITY conference was

> Wigderson, Naor, Tarjan, Leighton, Nisan,
> Raghavan, Yannakakis, Feige, Awerbuch, Galil,
> Yao, and Kosaraju.

In both cases, all constraints are satisfied and we observe that the committees are composed by well-known authorities in the fields. The running time for solving the IP in both cases is less than 1 second on a 3GHz Pentium 4 with 1GB memory, making the method very attractive to even larger datasets – for example, the corresponding IP for the IMDB dataset (containing hundreds of thousands variables in the constraints) is solved in 4min.

6 Conclusions

In this paper we introduce an approach to multi-relational data mining. The main idea is to find selectors that define projections on the data such that interesting patterns occur. We focus on datasets that consist of two relations that are connected into a chain. Patterns in this setting are expressed as graph properties. We show that many of the existing data mining problems can be cast as special cases of our framework, and we define a number of interesting novel data mining problems. We provide a characterization of properties for which one can apply level-wise methods. Additionally, we give an integer programming formulation of many interesting properties that allow us to solve the corresponding problems efficiently for medium-size instances of datasets in practice. In Table 1, the data mining problems we define in our framework are listed together with the property that defines them and the algorithmic tools we propose for their solution. Finally, we report experiments on two real datasets that demonstrate the benefits of our approach.

The current results are promising, but there are still many interesting questions on mining chains of relations. For example, the algorithmics of answering data mining queries on three-level graphs has many open problems. Level-wise search and other pattern discovery techniques provide efficient means to enumerate all feasible solutions for monotone and anti-monotone properties. However, the pattern discovery techniques are not limited to monotone and anti-monotone properties: it is sufficient that there is a relaxation of the property that is monotone or anti-monotone. Hence, finding monotone and anti-monotone relaxations

Problem	Property of G_S	Algorithmic tools
AUTHORITY(c) *	c has max degree in G_S	non-monotone, IP
BESTRANK(c)	$D_c^S \geq D_c^R$	non-monotone
CLIQUE	B_S bipartite clique	level-wise, IP
FREQUENCY(f, s)	B_S contains bipartite clique $K_{s,f\|P_S\|}$	non-monotone, IP association-rule mining
MAJORITY	every $a \in A_S$ has $\|E_{1,S}^a\| \geq \|E_1^a \setminus E_{1,S}^a\|$	non-monotone, IP
POPULARITY(b)	$\|A_S\| \geq b$	level-wise, IP
IMPACT(b)	for all $a \in A_S$, $D_a^S \geq b$	non-monotone, IP
ABSOLUTEIMPACT(b)	for all $a \in A_S$, $D_c \geq b$	level-wise, IP
COLLABORATIONCLIQUE	for every $a, b \in A_S$, at least one $p \in P_S$, s.t. $(a, p) \in E_{1,S}$ and $(b, p) \in E_{1,S}$	non-monotone, IP
CLASSIFICATION(c)	$P_S = \{p \in P : (c, p) \in E_1\}$ and $A_S = \{c\}$	non-monotone
PROGRAMCOMMITTEE(Z, l, m) *	$A_S = Z$, $\|S\| = m$, and every $t \in Z$ is connected to at least l nodes in S	IP

Table 1. Summary of problems and proposed algorithmic tools. Input is $G = (A, P, T; E_1, E_2)$. Given a selector set $S \subseteq T$ we have defined $G_S = (A_S, P_S, S; E_{1,S}, E_{2,S})$, and $B_S = (A_S, P_S; E_{1,S})$. By S we denote the selector set which is a solution and by R any selector set. D_c^S (D_c^R resp.) is the degree of c in G_S (G_R resp.) and D_c is the degree of c in G. The asterisk means that experiments are run on variants of these problems and also that these problems are discussed in more detail in this paper.

of the properties that are not monotone nor anti-monotone themselves is a potential direction of further research. Although many data mining queries on three-level graphs can be answered quite efficiently using off-the-shelf MILP solvers in practice for instances of moderate size, more sophisticated optimization techniques for particular mining queries, both in theory and in practice. Answering to multiple data mining queries on three-level graphs and updating the query answers when the graphs are interesting questions with practical relevance in data mining systems for chains of relations.

We have demonstrated the use of the framework using two datasets, but further experimental studies with the framework solving large-scale real-world data mining tasks would be of interest. We have done some preliminary studies on some biological datasets using the basic three-level framework. In real-world applications it would often be useful to extend the basic three-level graph framework in order to the actual data better into account. Extending the basic model to weighted edges, various interpretations, and more complex schemas seem a promising and relevant future direction in practice. There is a trade-off between the expressivity of the framework and the computational feasibility of the data mining queries. To cope with complex data, it would be very useful to have semi-automatic techniques to discover simple views to complex database schemas that capture relevant mining queries in our framework, in addition to generalizing our query answering techniques to more complex database schemas.

References

1. N. Agarwal, H. Liu, L. Tang, and P. S. Yu, "Identifying the influential bloggers in a community," in *WSDM*, 2008.
2. R. Agrawal, T. Imielinski, and A. N. Swami, "Mining association rules between sets of items in large databases," in *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, Washington, D.C., May 26-28, 1993*, P. Buneman and S. Jajodia, Eds. ACM Press, 1993, pp. 207–216.
3. M. Anthony and N. Biggs, *Computational Learning Theory: An Introduction*, paperback ed. Cambridge University Press, 1997.
4. L. Backstrom, D. P. Huttenlocher, J. M. Kleinberg, and X. Lan, "Group formation in large social networks: membership, growth, and evolution," in *KDD*, 2006, pp. 44–54.
5. R. J. Bayardo, B. Goethals, and M. J. Zaki, Eds., *FIMI '04, Proceedings of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations, Brighton, UK, November 1, 2004*, ser. CEUR Workshop Proceedings, vol. 126. CEUR-WS.org, 2004.
6. A. Borodin, G. O. Roberts, J. S. Rosenthal, and P. Tsaparas, "Link analysis ranking: Algorithms, theory, and experiments," in *ACM Transactions on Internet Technologies*, vol. 5, no. 1, Feb. 2005.
7. J.-F. Boulicaut, A. Bykowski, and C. Rigotti, "Free-sets: A condensed representation of boolean data for the approximation of frequency queries." *Data Mining and Knowledge Discovery*, vol. 7, no. 1, pp. 5–22, 2003.
8. T. Calders, L. V. S. Lakshmanan, R. T. Ng, and J. Paredaens, "Expressive power of an algebra for data mining," *ACM Trans. Database Syst.*, vol. 31, pp. 1169–1214, December 2006.

9. R. Caruana, "Multitask learning," *Machine Learning*, vol. 28, no. 1, pp. 41–75, 1997.
10. L. Cerf, J. Besson, C. Robardet, and J. franois Boulicaut, "Data peeler: Contraint-based closed pattern mining in n-ary relations," in *SIAM International Conference on Data Mining*, 2008, pp. 37–48.
11. L. Cerf, J. Besson, C. Robardet, and J.-F. Boulicaut, "Closed patterns meet n-ary relations," *ACM Trans. Knowl. Discov. Data*, vol. 3, pp. 3:1–3:36, March 2009.
12. W. Chen, C. Wang, and Y. Wang, "Scalable influence maximization for prevalent viral marketing in large-scale social networks," in *KDD*. ACM, 2010.
13. W. Chen, Y. Wang, and S. Yang, "Efficient influence maximization in social networks," in *KDD*, 2009.
14. V. Chvátal, "A greedy heuristic for the set-covering problem," *Mathematics of Operations Research*, vol. 4, no. 3, pp. 233–235, 1979.
15. A. Clare, H. E. Williams, and N. Lester, "Scalable multi-relational association mining," in *Proceedings of the 4th IEEE International Conference on Data Mining (ICDM 2004), 1-4 November 2004, Brighton, UK*. IEEE Computer Society, 2004, pp. 355–358.
16. D. J. Cook and L. B. Holder, "Graph-based data mining," *IEEE Intelligent Systems*, vol. 15, no. 2, pp. 32–41, 2000.
17. V. S. Costa, A. Srinivasan, R. Camacho, H. Blockeel, B. Demoen, G. Janssens, J. Struyf, H. Vandecasteele, and W. V. Laer, "Query transformations for improving the efficiency of ILP systems," *Journal of Machine Learning Research*, vol. 4, pp. 465–491, 2003.
18. L. Dehaspe and L. de Raedt, "Mining association rules in multiple relations," in *Inductive Logic Programming, 7th International Workshop, ILP-97, Prague, Czech Republic, September 17-20, 1997, Proceedings*, ser. Lecture Notes in Computer Science, N. Lavrac and S. Dzeroski, Eds., vol. 1297. Springer, 1997, pp. 125–132.
19. L. Dehaspe and H. Toivonen, "Discovery of frequent DATALOG patterns," *Data Mining and Knowledge Discovery*, vol. 3, no. 1, pp. 7–36, 1999.
20. H. Deng, M. R. Lyu, and I. King, "A generalized co-hits algorithm and its application to bipartite graphs," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. New York, NY, USA: ACM, 2009, pp. 239–248.
21. S. Dzeroski and N. Lavrac, Eds., *Relational Data Mining*. Springer, 2001.
22. R. Fagin, R. V. Guha, R. Kumar, J. Novak, D. Sivakumar, and A. Tomkins, "Multi-structural databases," in *Proceedings of the Twenty-fourth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, June 13-15, 2005, Baltimore, Maryland, USA*, C. Li, Ed. ACM, 2005, pp. 184–195.
23. U. Feige, "A threshold of ln n for approximating set cover," *Journal of the ACM*, vol. 45, no. 4, pp. 634–652, 1998.
24. G. C. Garriga, R. Khardon, and L. De Raedt, "On mining closed sets in multi-relational data," in *Proceedings of the 20th international joint conference on Artifical intelligence*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2007, pp. 804–809.
25. D. Gibson, J. M. Kleinberg, and P. Raghavan, "Inferring web communities from link topology," in *HYPERTEXT '98. Proceedings of the Ninth ACM Conference on Hypertext and Hypermedia: Links, Objects, Time and Space - Structure in Hypermedia Systems, June 20-24, 1998, Pittsburgh, PA, USA*. ACM, 1998, pp. 225–234.

26. A. Goyal, W. Lu, and L. V. Lakshmanan, "Celf++: optimizing the greedy algorithm for influence maximization in social networks," in *WWW*. ACM, 2011, pp. 47–48.
27. D. Haussler, "Quantifying inductive bias: AI learning algorithms and Valiant's learning framework," *Artificial Intelligence*, vol. 36, no. 2, pp. 177–221, 1988.
28. T. Horváth, "Cyclic pattern kernels revisited," in *Advances in Knowledge Discovery and Data Mining, 9th Pacific-Asia Conference, PAKDD 2005, Hanoi, Vietnam, May 18-20, 2005, Proceedings*, ser. Lecture Notes in Computer Science, T. B. Ho, D. Cheung, and H. Liu, Eds., vol. 3518. Springer, 2005, pp. 791–801.
29. T. Horváth, T. Gärtner, and S. Wrobel, "Cyclic pattern kernels for predictive graph mining," in *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, Washington, USA, August 22-25, 2004*, W. Kim, R. Kohavi, J. Gehrke, and W. DuMouchel, Eds. ACM, 2004, pp. 158–167.
30. J. Huan, W. Wang, and J. Prins, "Efficient mining of frequent subgraphs in the presence of isomorphism," in *Proceedings of the 3rd IEEE International Conference on Data Mining (ICDM 2003), 19-22 December 2003, Melbourne, Florida, USA*. IEEE Computer Society, 2003, pp. 549–552.
31. J. Huan, W. Wang, J. Prins, and J. Yang, "SPIN: mining maximal frequent subgraphs from graph databases," in *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, Washington, USA, August 22-25, 2004*, W. Kim, R. Kohavi, J. Gehrke, and W. DuMouchel, Eds. ACM, 2004, pp. 581–586.
32. R. Jaschke, A. Hotho, C. Schmitz, B. Ganter, and S. Gerd, "Trias–an algorithm for mining iceberg tri-lattices," in *Proceedings of the Sixth International Conference on Data Mining*. Washington, DC, USA: IEEE Computer Society, 2006, pp. 907–911.
33. G. Jeh and J. Widom, "Mining the space of graph properties," in *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, Washington, USA, August 22-25, 2004*, W. Kim, R. Kohavi, J. Gehrke, and W. DuMouchel, Eds. ACM, 2004, pp. 187–196.
34. L. Ji, K.-L. Tan, and A. K. H. Tung, "Mining frequent closed cubes in 3d datasets," in *Proceedings of the 32nd international conference on Very large data bases*. VLDB Endowment, 2006, pp. 811–822.
35. Y. Jin, T. M. Murali, and N. Ramakrishnan, "Compositional mining of multirelational biological datasets," *ACM Trans. Knowl. Discov. Data*, vol. 2, pp. 2:1–2:35, April 2008.
36. U. Kang, C. E. Tsourakakis, and C. Faloutsos, "Pegasus: A peta-scale graph mining system," in *ICDM*, 2009, pp. 229–238.
37. D. Kempe, J. M. Kleinberg, and É. Tardos, "Maximizing the spread of influence through a social network," in *KDD*, 2003.
38. ——, "Influential nodes in a diffusion model for social networks," in *ICALP*, 2005.
39. R. Kumar, P. Raghavan, S. Rajagopalan, D. Sivakumar, A. Tomkins, and E. Upfal, "The web as a graph," in *Proceedings of the Nineteenth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, May 15-17, 2000, Dallas, Texas, USA*. ACM, 2000, pp. 1–10.
40. R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins, "Trawling the web for emerging cyber-communities," *Computer Networks*, vol. 31, no. 11-16, pp. 1481–1493, 1999.
41. M. Kuramochi and G. Karypis, "Frequent subgraph discovery," in *Proceedings of the 2001 IEEE International Conference on Data Mining, 29 November - 2*

December 2001, San Jose, California, USA, N. Cercone, T. Y. Lin, and X. Wu, Eds. IEEE Computer Society, 2001, pp. 313–320.
42. T. Lappas, K. Liu, and E. Terzi, "Finding a team of experts in social networks," in *KDD*, 2009.
43. T. Lappas, E. Terzi, D. Gunopulos, and H. Mannila, "Finding effectors in social networks," in *KDD*, 2010.
44. J. Leskovec, K. J. Lang, and M. W. Mahoney, "Empirical comparison of algorithms for network community detection," in *WWW*, 2010.
45. B. Long, X. Wu, Z. Zhang, and P. S. Yu, "Unsupervised learning on k-partite graphs," in *Knowledge Discovery and Data Mining*, 2006, pp. 317–326.
46. H. Mannila and E. Terzi, "Finding links and initiators: A graph-reconstruction problem," in *SDM*, 2009.
47. H. Mannila and H. Toivonen, "Levelwise search and borders of theories in knowledge discovery," *Data Minining and Knowledge Discovery*, vol. 1, no. 3, pp. 241–258, 1997.
48. A. Martin, "General mixed integer programming: Computational issues for branch-and-cut algorithms." in *Computational Combinatorial Optimization*, 2001, pp. 1–25.
49. M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Phys. Rev. E*, vol. 69, no. 2, p. 026113, Feb 2004.
50. L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank citation ranking: Bringing order to the web," Stanford University, Tech. Rep., 1998.
51. G. Pandurangan, P. Raghavan, and E. Upfal, "Using PageRank to characterize web structure," in *Computing and Combinatorics, 8th Annual International Conference, COCOON 2002, Singapore, August 15-17, 2002, Proceedings*, ser. Lecture Notes in Computer Science, O. H. Ibarra and L. Zhang, Eds., vol. 2387. Springer, 2002, pp. 330–339.
52. S. Rajagopalan and V. V. Vazirani, "Primal-dual RNC approximation algorithms for set cover and covering integer programs," *SIAM Journal on Computing*, vol. 28, no. 2, pp. 525–540, 1998.
53. S. Sarawagi and G. Sathe, "i^3: Intelligent, interactive investigation of OLAP data cubes," in *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, May 16-18, 2000, Dallas, Texas, USA*, W. Chen, J. F. Naughton, and P. A. Bernstein, Eds. ACM, 2000, p. 589.
54. L. Theodoros, L. Kun, and T. Evimaria, "A survey of algorithms and systems for expert location in social networks," in *Social Network Data Analytics*, C. C. Aggarwal, Ed. Springer US, 2011, pp. 215–241.
55. H. Tong, S. Papadimitriou, J. Sun, P. S. Yu, and C. Faloutsos, "Colibri: fast mining of large static and dynamic graphs," in *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2008.
56. C. Wang, W. Wang, J. Pei, Y. Zhu, and B. Shi, "Scalable mining of large disk-based graph databases," in *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, Washington, USA, August 22-25, 2004*, W. Kim, R. Kohavi, J. Gehrke, and W. DuMouchel, Eds. ACM, 2004, pp. 316–325.
57. T. Washio and H. Motoda, "State of the art of graph-based data mining," *SIGKDD Explorations*, vol. 5, no. 1, pp. 59–68, 2003.
58. X. Yan and J. Han, "Closegraph: mining closed frequent graph patterns," in *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 24 - 27, 2003*, L. Getoor, T. E. Senator, P. Domingos, and C. Faloutsos, Eds. ACM, 2003, pp. 286–295.

59. X. Yan, P. S. Yu, and J. Han, "Graph indexing: A frequent structure-based approach," in *Proceedings of the ACM SIGMOD International Conference on Management of Data, Paris, France, June 13-18, 2004*, G. Weikum, A. C. König, and S. Deßloch, Eds. ACM, 2004, pp. 335–346.
60. M. Yannakakis, "Node-and edge-deletion NP-complete problems," in *Proceedings of the tenth annual ACM symposium on Theory of computing, May 01-03, 1978, San Diego, California, United States*, R. J. Lipton, W. Burkhard, W. Savitch, E. P. Friedman, and A. Aho, Eds. ACM, 1978, pp. 253–264.
61. M. J. Zaki, "Efficiently mining frequent trees in a forest," in *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, July 23-26, 2002, Edmonton, Alberta, Canada.* ACM, 2002, pp. 71–80.
62. A. X. Zheng, A. Y. Ng, and M. I. Jordan, "Stable algorithms for link analysis," in *SIGIR 2001: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, September 9-13, 2001, New Orleans, Louisiana, USA*, W. B. Croft, D. J. Harper, D. H. Kraft, and J. Zobel, Eds. ACM, 2001, pp. 258–266.
63. Z. Zou, H. Gao, and J. Li, "Discovering frequent subgraphs over uncertain graph databases under probabilistic semantics," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining.* New York, NY, USA: ACM, 2010, pp. 633–642.