Mining chains of relations

Foto Afrati[*]
National Technical University of Athens,
Athens, Greece

Gautam Das[*]
University of Texas at Arlington,
Arlington, TX, USA

Aristides Gionis, Heikki Mannila, Taneli Mielikäinen, and Panayiotis Tsaparas
Basic Research Unit, HIIT, University of Helsinki, Helsinki, Finland

Abstract

Traditional data mining applications consider the problem of mining a single relation between two attributes. For example, in a scientific bibliography database, authors are related to papers, and we may be interested in discovering association rules between authors. However, in real life, we often have multiple attributes related though chains *of relations. For example, authors write papers, and papers concern one or more topics. Mining such relational chains poses additional challenges. In this paper we consider the following problem: given a chain of two relations $R_1(A, P)$ and $R_2(P, T)$ we want to find selectors for the objects in T such that the projected relation between A and P satisfies a specific property. The motivation for our approach is that a given property might not hold on the whole dataset, but it might hold when projecting the data on a selector set. We discuss various algorithms and we examine the conditions under which the apriori technique can be used. We experimentally demonstrate the effectiveness of our methods.*

1 Introduction

Traditional data mining applications extract interesting patterns from a single relation between two attributes, e.g., customers buying products, documents containing words, or genes expressed in tissues. Multi-relational data mining has been considered an extension to the simple transactional data model. However, addressing the problem in the full generality proved to be a daunting task. In this paper, we restrict our attention to a specific case of chains of relations. Our formulation captures many practical applications, yet the problems are still easy to formulate and feasible to solve.

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 a and b frequently write papers together" might not be true for the whole dataset, but it might be true for a specific topic t. Therefore, it is meaningful to search for projections of the data on which interesting patterns occur. We model datasets as graphs and patterns as graph properties. Under this light, the data-mining problem is to find nodes (selectors to project the data on) so that the induced subgraph (the projected data) satisfies a given property.

In the most general case of a database schema we assume n attributes $A_1, \ldots, A_n$, and m relations $R_1, \ldots, R_m$ on the attributes. In this paper, we focus on a simple scheme with three attributes A, B and C, and a chain of two relations $R_1(A, B)$ and $R_2(B, C)$. However, the general ideas of our extension can be applied to more complex schemas.

We find it convenient to work with the graph representation of schemas:[1] we assume a graph G 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 has two sets of edges, E_1 connecting nodes in A and B, and E_2 connecting nodes in B and C. Edges are defined so that $(a, b) \in E_1$ if and only if $(a, b) \in R_1$, and $(b, c) \in E_2$ if and only if $(b, c) \in R_2$. We call such a graph a *three-level graph*. Examples of datasets that can be modeled with three-level graphs include: AUTHORS writing PAPERS about TOPICS; ACTORS playing in MOVIES belonging to GENRES; and DOCUMENTS containing PARAGRAPHS containing WORDS.

The general data-mining problem we consider is informally defined as follows. Consider the three-level graph $G = (A, B, C; E_1, E_2)$. 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 B' is connected to a node in C' with an edge in E_2, and every node

[*] This work was done while the author was visiting Basic Research Unit, HIIT, University of Helsinki, Helsinki Finland

[1] 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.

in A' is connected to a node in B' with an edge in E_1. Now the induced subgraph G' might satisfy a given property or it might not. Example of properties are "G' contains a bipartite clique $K_{s,t}$", and "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. Note that many traditional data-mining problems can be viewed as finding subgraphs with certain properties. For instance, finding itemsets of size s and support t in market-basket data corresponds to finding $K_{s,t}$ cliques.

For solving the general problem, we provide conditions under which monotonicity properties hold, and thus, a level-wise method like apriori (see, e.g., [8]) can be used. Many of the problems we consider are NP-hard — many of them are hard instances of node removal problems [11]. For such problems we propose an Integer Programming (IP) formulation that can be used to solve medium-size instances by using existing IP solvers.

2 Problem definition

We start with attributes A, P and T, relations $E_1(A,P)$ and $E_2(P,T)$, and the corresponding three-level graph $G = (A,P,T;E_1,E_2)$.[2] Let $S \subseteq T$ be a subset of T. The set S acts as a selector over the sets P and A. Namely, for some $t \in T$, we define $P_t = \{p \in P : (p,t) \in E_2\}$, and $A_t = \{a \in A : \exists p \in P_t, \text{ s.t. } (a,p) \in E_1\}$. That is, the sets A_t and P_t are the subsets of nodes in P and A, respectively, that are reachable from the node $t \in T$. We can extend the definition to subsets P_S and A_S that are reachable from the set $S \subseteq T$. Extending the definition to sets requires to define the *interpretation* $\mathcal{I}$ of the selector S.

Disjunctive Interpretation ($\mathcal{D}$): the sets P_S and A_S are reachable from *at least* one node in S:

$$P_S^{\mathcal{D}} = \bigcup_{t \in T} P_t \quad \text{and} \quad A_S^{\mathcal{D}} = \bigcup_{t \in T} A_t.$$

Conjunctive Interpretation ($\mathcal{C}$): the sets P_S and A_S are reachable from *every* node in S:

$$P_S^{\mathcal{C}} = \bigcap_{t \in T} P_t \quad \text{and} \quad A_S^{\mathcal{C}} = \bigcap_{t \in T} A_t.$$

Given the subsets A_S and P_S we can define the *induced* bipartite subgraph $G_S = (A_S, P_S; E_S)$, where $E_S = \{(a,p) \in E_1 : a \in A_S, p \in P_S\}$. We use $G_S^{\mathcal{D}}$ and $G_S^{\mathcal{C}}$ to denote the induced subgraph under disjunctive and conjunctive interpretation respectively.

Now let Ψ denote a graph property. Given a three-level graph G, a property Ψ, and the interpretation $\mathcal{I}$, we define the following problems.

MAX$_{\mathcal{I}}$-$\Psi(G)$**:** Find the maximal set $S \subseteq T$ such that $G_S^{\mathcal{I}}$ satisfies Ψ.

MIN$_{\mathcal{I}}$-$\Psi(G)$**:** Find the minimal set $S \subseteq T$ such that $G_S^{\mathcal{I}}$ satisfies Ψ.

ANY$_{\mathcal{I}}$-$\Psi(G)$**:** Find any set $S \subseteq T$ such that $G_S^{\mathcal{I}}$ satisfies Ψ.

Below we give a few examples of interesting properties. Many more can be found in the full version of the paper. In all cases, we the graph $G = (A,P,T;E_1,E_2)$ is the input, and $G_S = (A_S,P_S;E_S)$, is the graph induced by the selector S, under either interpretation.

AUTHORITY(c): Given a node $c \in A$, the graph G_S satisfies AUTHORITY(c) if $c \in A_S$, and c has the maximum degree among all nodes in A_S.

CLIQUE: The graph G_S satisfies CLIQUE if it is a bi-clique.

FREQUENCY(f): Given threshold value $f \in [0,1]$, the graph G_S satisfies the property FREQUENCY(f) if G_S contains a bipartite clique $K_{s,f|P_S|}$ for some $s > 0$. Also interesting is to enumerate all such cliques. The intuition is that a bipartite clique $K_{s,f|P_S|}$ implies a frequent itemset of size s with frequency threshold f.

PROGRAMCOMMITTEE(Z,l,m): We are given a set $Z \subseteq T$ (topics of a conference), and values l and m. We say that the induced subgraph $B = (X,Y,Z;E,F)$ satisfies the property PROGRAMCOMMITTEE(Z,l,m) if $|X| = m$ (m members in the program committee) and every node $t \in Z$ is connected to at least l nodes in X (for each topic there are at least l experts in the committee). Notice that this is the only example in which we reverse the roles of the attributes A and T, that is, the input specifies a subsets of nodes from T and the selector set is chosen from A. However, there is no real conceptual difference with the other examples, we make only this exception to be consistent with the semantics of the bibliography dataset.

3 A characterization of monotonicity

We now characterize graph properties for which standard level-wise methods can be used. Given a three-level graph G and an interpretation $\mathcal{I} \in \{\mathcal{C},\mathcal{D}\}$, let $\mathcal{S}^{\mathcal{I}} = \{G_S^{\mathcal{I}} : S \subseteq T\}$ be the set of all possible induced bipartite graphs under interpretation $\mathcal{I}$.

Definition 1 *A property Ψ is* monotone *(*anti-monotone*) on the set $\mathcal{S}^{\mathcal{I}}$ if for any selector set $S \subseteq T$ such that $G_S^{\mathcal{I}}$ satisfies Ψ, we have that $G_{S'}^{\mathcal{I}}$ satisfies Ψ for all $S' \subseteq S$ (for all $S' \supseteq S$).*

Monotonicity is used to prune the search space, by not considering supersets of a selector set that does not satisfy a

[2] The attribute naming and the names of the problems and the graph properties we introduce later are inspired by the bibliography dataset (AUTHORS – PAPERS – TOPICS).

property [1]. Here, we relate monotonicity with the concept of *hereditary* properties on graphs.

Definition 2 *A property Ψ is* hereditary *on the set $\mathcal{G}$ of all possible graphs with respect to node deletions, if the following is true. If $G = (V, E)$ is a graph that satisfies Ψ, then for any $V' \subseteq V$ the induced subgraph $G' = (V', E')$ of G also satisfies the property.*

We can prove the following. We omit the proof due to lack of space.

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

Theorem 1 has many interesting consequences, e.g.,

Proposition 1 *The* CLIQUE *property is monotone on $\mathcal{S}^{\mathcal{D}}$.*

4 Integer Programming formulations

Computing the *maximal* or the *minimal* or *any* selector set in many cases is an NP-hard problem. Here we give IP formulations for such problems. 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.[3] For the following we assume a disjunctive interpretation, but similar formulations can be given for the conjunctive case.

Let $G = (A, P, T, E_1, E_2)$ be a three-level graph. 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 0 otherwise. Similarly, for each element $j \in P$ we define a variable $p_j \in \{0, 1\}$. First we require that if an element $i \in T$ is chosen, then the set $P_i^T = \{j \in P : (j, i) \in E_2\}$ is also chosen. This condition is enforced by requiring that

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

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 $T_j^P = \{i \in T : (j, i) \in E_2\}$. We have that

$$\sum_{i \in T_j^P} t_i \geq p_j.$$

Finally, for each element $k \in A$, we define a variable $a_k \in \{0, 1\}$ and impose similar constraints. Let $A_j^P = \{k : (k, j) \in E_1\}$ and $P_k^A = \{j : (k, j) \in E_1\}$. Then we have

$$a_k \geq p_j \text{ for all } k \in A_j^P, \text{ and } \sum_{j \in P_k^A} p_j \geq a_k.$$

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., $x_k = \sum_{j \in P_k^A} p_j$.

For the properties we discussed in Section 2 we need to impose additional restrictions on the different variables.

AUTHORITY(c)**:** We require $x_c \geq x_k$ for all $k \in A - \{c\}$.

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 $\bigcup_{i \in Z} A_i^T$.

5 Algorithmic considerations

The FREQUENCY **problem:** We can show that finding frequent itemsets V with frequency threshold f in the three-level graphs is equivalent to finding *association rules* $S \rightarrow V$ with confidence threshold f. We omit the details.

The AUTHORITY **problem:** For a single author c, we solve the authority problem using MIP. The actual details depend on which interpretation we use.

In the conjunctive interpretation, the subgraph induced by a topic set S contains a paper $j \in P$ iff $S \subseteq T_j^P$. Thus, we can consider each paper $j \in P$ as a topic set T_j^P. Finding all topic sets with nonempty induced subgraph corresponds to mining frequent sets with frequency threshold $f = 1/|P|$ in the database consisting the topic sets T_j^P, and frequent set mining algorithms can be used, e.g., see [1].

In the disjunctive interpretation, the subgraph induced by the topic set S contains a paper $j \in P$ iff 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. It can be to shown that such sets of topics correspond to minimal hypergraph transversals and their subsets in the hypergraph $(T, \{T_j^P\}_{j \in P})$. They can be generated efficiently by the level-wise approach.

The PROGRAMCOMMITTEE **problem:** For the PROGRAMCOMMITTEE problem we use the MIP formulation sketched in Section 4. For an objective function, we use the number of papers written by the program committee about the topics in the given set Z.

6 Experiments

We used information available on the Web to construct two real datasets with three-level structure. For the datasets we used we found more interesting to experiment with the AUTHORITY and the PROGRAMCOMMITTEE problem.

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

Bibliography datasets: We crawled the ACM digital library website[4] and we extracted information about two forums: Journal of ACM (JACM) and ACM Symposium on Theory of Computing (STOC). For each paper we obtained the authors, the title, and the topics. Examples of topics are "numerical analysis", "programming languages", and "discrete mathematics". In the JACM dataset we have 2112 authors, 2321 papers and 56 topics. In the STOC dataset we have 1404 authors, 1790 papers and 48 topics.

IMDB dataset: We use IMDB[5] to extract an actors-movies-genres dataset. We preprocess the original dataset to prune TV serials, non-English movies, movies with no genre, as well as actors with secondary roles. We have 45342 actors, 71912 movies and 21 genres.

The AUTHORITY **problem:** We run the level-wise algorithms described in Section 5 on our three datasets. Given an author a, we define the collection of topic sets $\mathcal{A}(a) = \{S : a \text{ is authority for topic set } S\}$, and $\mathcal{A}^0(a)$ the collection of minimal sets of $\mathcal{A}(a)$, or more precisely $\mathcal{A}^0(a) = \{S \in \mathcal{A}(a) : \nexists T \in \mathcal{A}(a), \text{ with } T \subset S\}$. For authors who are not authorities, $\mathcal{A}(a)$ and $\mathcal{A}^0(a)$ are empty.

The author with the most papers in STOC is Wigderson (36 papers). 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 (25 papers), has corresponding values 2 and 1.5. That is, he is very focused on two topic sets: "data structures" and ("discrete mathematics", "artificial intelligence"). These indicative results match our intuition about the authors.

We also searched for authorities in the JACM and IMDB datasets, but we omit the results due to lack of space. 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.

The PROGRAMCOMMITTEE **problem:** The task is to select PC members for a subset of topics (potential conference). We give two examples of selecting PC members for two fictional conferences. For the first conference, which we called LOGIC-AI, we used 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 the topics "discrete mathematics", "analysis of algorithms and problem complexity", "computation by abstract devices", and "data structures". We requested 12-member committees requiring each topic to be covered by at least 4 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 the committees are composed by well-known authorities in the fields.

7 Related work

There has already been some effort on *multi-relational mining* [2, 3, 4]. The approach taken 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 [9] and with the very recent multi-structural databases [5]. Our work on mining layered graphs also has connections with the general area of *graph mining*, in which various problems have been investigated, ranging from mining frequent subgraphs [7, 10, 12] to extraction of web communities [6], and so on.

References

[1] R. Bayardo, B. Goethals, and M. J. Zaki, editors. *Proceedings of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations*, 2004.

[2] A. Clare, H. E. Williams, and N. Lester. Scalable multi-relational association mining. In *ICDM*, 2004.

[3] L. Dehaspe and L. D. Raedt. Mining association rules in multiple relations. In *ILP*, 1997.

[4] S. Dzeroski and N. L. editors. *Relational Data Mining*. Springer, 2001.

[5] R. Fagin, R. Guha, R. Kumar, J. Novak, D. Sivakumar, and A. Tomkins. Multi-structural databases. In *PODS*, 2005.

[6] D. Gibson, J. Kleinberg, and P. Raghavan. Inferring web communities from link topology. In *HYPERTEXT*, 1998.

[7] M. Kuramochi and G. Karypis. Frequent subgraph discovery. In *ICDM*, 2001.

[8] H. Mannila and H. Toivonen. Levelwise search and borders of theories in knowledge discovery. *Data Mining and Knowledge Discovery*, 1(3):241–258, 1997.

[9] S. Sarawagi and G. Sathe. i^3: Intelligent, interactive investigaton of olap data cubes. In *SIGMOD*, 2000.

[10] X. Yan, P. S. Yu, and J. Han. Graph indexing: a frequent structure-based approach. In *SIGMOD*, 2004.

[11] M. Yannakakis. Node-and edge-deletion NP-complete problems. In *STOC*, 1978.

[12] M. J. Zaki. Efficiently mining frequent trees in a forest. In *KDD*, 2002.

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

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