Discovering Interesting Patterns in Large Graph Cubes

Florian Demesmaeker*†, Amine Ghrab†, Siegfried Nijssen* and Sabri Skhiri†

*Université Catholique de Louvain, Louvain-La-Neuve, Belgium
Email: siegfried.nijssen@uclouvain.be
†EURANOVA R&D, Mont-Saint-Guibert, Belgium
Email: firstname.lastname@euranova.eu

Abstract—Due to the increasing importance and volume of highly interconnected data, such as in social or information networks, a plethora of graph mining techniques have been designed to enable the analysis of such data. In this work, we focus on the mining of associations between entity features in networks. We model each entity feature as a dimension to be analyzed. Consequently, we build our approach on top of the existing graph cube framework which is an extension of the concept of the data cube to networks. Our task is particularly challenging because it requires the analysis of both the initial multidimensional network and all its subsequent aggregate forms. As soon as we deal with a big data situation it is impossible for an analyst to consider manually all the possible views of the network data. The aim of this work is to design an algorithm for the discovery of interesting patterns in large graph cubes. Thus, instead of examining all the possible aggregations manually, the proposed technique leads the analyst to the interesting associations or patterns in the multidimensional network. Furthermore, we study the application of existing algorithms from the frequent itemset mining literature on graph data and propose a mapping between the two settings.

Keywords—Graph Mining, Graph Cubes, Frequent Itemset Mining

I. INTRODUCTION

Due to the increasing availability of network data, various algorithms are being developed by the data mining community for the automated analysis of graphs. Moreover, richer network information has become accessible, creating the need for graph mining techniques that consider both the network structure and the entities’ features. In this work, we search for interesting patterns in large graphs. Among the different kinds of interesting patterns we may find, we focus on surprising patterns. For example, let us say that we observe a high number of relationships between American and Chinese people. This is a pattern we may find in a social network. However, since a large number of people live in the USA and China, it is not surprising that the number of relationships between people of these two countries is significant. On the other hand, observing the same high number of connections between Belgian and Chinese people is more surprising. In this work, we are looking for such surprising patterns over the characteristics of edges. To measure how surprising a pattern is, we define a null model that we assume has generated the data at hand.

We model the graph of interest by a multidimensional network where nodes have attributes while edges have not. The graph cube framework proposed by Zhao et al. [1] is an extension of the data cube to graph data. From a multidimensional network this framework defines the different network aggregations, which are views of the original network without one or multiple attributes. In such an aggregate network nodes are merged in a way similar to a GROUP BY in a relational database and the weights of the edges are updated. Example 1 presents a multidimensional network and an aggregate network built from it.

Example 1: Figure 1a presents our toy network that contains 10 nodes and 13 edges. Each node represents a person whose gender, location and profession are known. The location attribute can take three different values: Belgium (BE), The Netherlands (NL) or Germany (DE). Then the profession value is one of the following: Lawyer, Engineer or Teacher.

In this network multiple patterns along one or multiple attributes can be found. For instance, we can consider [(Engineer), (Teacher)] or [(Lawyer, Lawyer)] that are two patterns defined over the Profession attribute only. But we can also be more specific by considering a pattern defined over the Location and Profession attributes, e.g. [(Belgium, Teacher), (Germany, Engineer)]. The two aggregations considered here are (Profession) and (Location, Profession) respectively. Figure 1b presents the aggregate network (Profession) where the location and the gender of the people are ignored. Section III introduces formally the graph cube and its aggregate networks.

A graph cube represents all the possible aggregations of the initial multidimensional network. Graph cube mining is then no longer limited to the initial graph, but also explores all possible aggregations of the graph. Interesting patterns could be discovered inside each aggregate network. Each aggregation is built by considering or omitting every dimension of the original network. As a consequence, the graph cube contains an exponential number of aggregate networks. In a big data setting, that is where the number of dimensions or the network itself is large, an analyst is overwhelmed by the number of patterns. Hence, we propose an algorithm to help the analyst locate the interesting combinations of dimensions as well as the interesting patterns in these views.
The algorithms proposed in the state of the art allow the discovery of interesting patterns in a single graph mainly by analyzing its structure. In this work we consider both the structure of the graph and the internal information present in each node. We employ a statistical technique inspired from the frequent itemset mining literature to locate interesting patterns in large graph cubes. Thus, instead of examining the large cubes space manually, the proposed technique leads the analyst to the surprising patterns of the large multidimensional network.

Existing algorithms that discover interesting itemsets include MINI [2] and MTV [3]. MINI searches for surprising itemsets while MTV searches for representative itemsets. As MINI, our approach searches for surprising associations. Therefore we propose a mapping between pattern mining in attributed graphs and frequent itemset mining in transactional databases.

The algorithms proposed in the state of the art allow the discovery of interesting patterns in a single graph mainly by analyzing its structure. In this work we consider both the structure of the graph and the internal information present in each node. We employ a statistical technique inspired from the frequent itemset mining literature to locate interesting patterns in large graph cubes. Thus, instead of examining the large cubes space manually, the proposed technique leads the analyst to the surprising patterns of the large multidimensional network.

The algorithms proposed in the state of the art allow the discovery of interesting patterns in a single graph mainly by analyzing its structure. In this work we consider both the structure of the graph and the internal information present in each node. We employ a statistical technique inspired from the frequent itemset mining literature to locate interesting patterns in large graph cubes. Thus, instead of examining the large cubes space manually, the proposed technique leads the analyst to the surprising patterns of the large multidimensional network.

The algorithms proposed in the state of the art allow the discovery of interesting patterns in a single graph mainly by analyzing its structure. In this work we consider both the structure of the graph and the internal information present in each node. We employ a statistical technique inspired from the frequent itemset mining literature to locate interesting patterns in large graph cubes. Thus, instead of examining the large cubes space manually, the proposed technique leads the analyst to the surprising patterns of the large multidimensional network.

The algorithms proposed in the state of the art allow the discovery of interesting patterns in a single graph mainly by analyzing its structure. In this work we consider both the structure of the graph and the internal information present in each node. We employ a statistical technique inspired from the frequent itemset mining literature to locate interesting patterns in large graph cubes. Thus, instead of examining the large cubes space manually, the proposed technique leads the analyst to the surprising patterns of the large multidimensional network.

The algorithms proposed in the state of the art allow the discovery of interesting patterns in a single graph mainly by analyzing its structure. In this work we consider both the structure of the graph and the internal information present in each node. We employ a statistical technique inspired from the frequent itemset mining literature to locate interesting patterns in large graph cubes. Thus, instead of examining the large cubes space manually, the proposed technique leads the analyst to the surprising patterns of the large multidimensional network.

The algorithms proposed in the state of the art allow the discovery of interesting patterns in a single graph mainly by analyzing its structure. In this work we consider both the structure of the graph and the internal information present in each node. We employ a statistical technique inspired from the frequent itemset mining literature to locate interesting patterns in large graph cubes. Thus, instead of examining the large cubes space manually, the proposed technique leads the analyst to the surprising patterns of the large multidimensional network.

The algorithms proposed in the state of the art allow the discovery of interesting patterns in a single graph mainly by analyzing its structure. In this work we consider both the structure of the graph and the internal information present in each node. We employ a statistical technique inspired from the frequent itemset mining literature to locate interesting patterns in large graph cubes. Thus, instead of examining the large cubes space manually, the proposed technique leads the analyst to the surprising patterns of the large multidimensional network.

The algorithms proposed in the state of the art allow the discovery of interesting patterns in a single graph mainly by analyzing its structure. In this work we consider both the structure of the graph and the internal information present in each node. We employ a statistical technique inspired from the frequent itemset mining literature to locate interesting patterns in large graph cubes. Thus, instead of examining the large cubes space manually, the proposed technique leads the analyst to the surprising patterns of the large multidimensional network.

The algorithms proposed in the state of the art allow the discovery of interesting patterns in a single graph mainly by analyzing its structure. In this work we consider both the structure of the graph and the internal information present in each node. We employ a statistical technique inspired from the frequent itemset mining literature to locate interesting patterns in large graph cubes. Thus, instead of examining the large cubes space manually, the proposed technique leads the analyst to the surprising patterns of the large multidimensional network.

The algorithms proposed in the state of the art allow the discovery of interesting patterns in a single graph mainly by analyzing its structure. In this work we consider both the structure of the graph and the internal information present in each node. We employ a statistical technique inspired from the frequent itemset mining literature to locate interesting patterns in large graph cubes. Thus, instead of examining the large cubes space manually, the proposed technique leads the analyst to the surprising patterns of the large multidimensional network.

The algorithms proposed in the state of the art allow the discovery of interesting patterns in a single graph mainly by analyzing its structure. In this work we consider both the structure of the graph and the internal information present in each node. We employ a statistical technique inspired from the frequent itemset mining literature to locate interesting patterns in large graph cubes. Thus, instead of examining the large cubes space manually, the proposed technique leads the analyst to the surprising patterns of the large multidimensional network.

The algorithms proposed in the state of the art allow the discovery of interesting patterns in a single graph mainly by analyzing its structure. In this work we consider both the structure of the graph and the internal information present in each node. We employ a statistical technique inspired from the frequent itemset mining literature to locate interesting patterns in large graph cubes. Thus, instead of examining the large cubes space manually, the proposed technique leads the analyst to the surprising patterns of the large multidimensional network.

The algorithms proposed in the state of the art allow the discovery of interesting patterns in a single graph mainly by analyzing its structure. In this work we consider both the structure of the graph and the internal information present in each node. We employ a statistical technique inspired from the frequent itemset mining literature to locate interesting patterns in large graph cubes. Thus, instead of examining the large cubes space manually, the proposed technique leads the analyst to the surprising patterns of the large multidimensional network.

The algorithms proposed in the state of the art allow the discovery of interesting patterns in a single graph mainly by analyzing its structure. In this work we consider both the structure of the graph and the internal information present in each node. We employ a statistical technique inspired from the frequent itemset mining literature to locate interesting patterns in large graph cubes. Thus, instead of examining the large cubes space manually, the proposed technique leads the analyst to the surprising patterns of the large multidimensional network.

The algorithms proposed in the state of the art allow the discovery of interesting patterns in a single graph mainly by analyzing its structure. In this work we consider both the structure of the graph and the internal information present in each node. We employ a statistical technique inspired from the frequent itemset mining literature to locate interesting patterns in large graph cubes. Thus, instead of examining the large cubes space manually, the proposed technique leads the analyst to the surprising patterns of the large multidimensional network.

In the remainder of this section we briefly present the frequent itemset mining setting and the MINI algorithm that searches for surprising itemsets. As we shall see, sets of attributes in a network can be seen as itemsets. Then we present another approach and the MTV algorithm that are based on the maximum entropy principle. This approach searches for itemsets that best describe the data and therefore is opposed to the hypothesis testing framework that we propose. To end this section, we discuss an approach whose goal is highly similar to ours. It was recently proposed to find the most interesting aggregate networks in a graph cube.

Frequent itemset mining: The mining model, widely used in the frequent itemset literature, was first introduced by Agrawal et al. [10]. Given a set of transactions, each containing a set of items, the task of a frequent itemset miner is to discover items that occur together in many transactions, and to build association rules. The number of transactions in which an itemset \( I \) is present is called the support of \( I \). Since the number of subsets is exponential with its size, a prohibitively large number of frequent itemsets could be found. Therefore, an interesting phenomenon, called the pattern explosion, arises with frequent itemset mining. Efficient algorithms were designed to mine frequent itemsets on large data. Agrawal and Srikant presented the Apriori algorithm [11] to perform such a task. Moreover, several condensed representations of the frequent itemsets have been

**Surprising itemsets:** However, an itemset that is frequent is not necessarily interesting. Gallo et al. presented MINI [2], an algorithm reporting the Most Informative and Non-redundant Itemsets. The idea behind MINI is to select itemsets that are hard to be explained by chance under prior knowledge on the data. Itemsets are qualified as informative if they are surprising with respect to some prior knowledge on the data. Moreover MINI ensures that overlapping itemsets are not both highly ranked as informative as they convey part of the same information.

**Maximum entropy models and subjective interestingness:** De Bie [15] presented a generic framework to represent prior information using the maximum entropy principle [16] by a probability distribution called MaxEnt. The entropy in the sense of Shannon [17] is a measure of uncertainty about a state. It can also be interpreted as the average quantity of information in this state or the number of bits needed to encode such information. Once the prior information has been used to build a probability distribution on the data, it is possible to evaluate the interestingness of a pattern.

The MaxEnt model can be used to quantify the interestingness of a pattern, for instance by considering its probability under MaxEnt. The negative log-probability is known as the self-information [18] in Shannon’s information theory, the larger the more informative.

**Succinctly summarizing data with itemsets:** Mampaey et al. presented the MTV algorithm [3] to summarize data with itemsets. Their work is based on the generic framework of De Bie [15] to model the prior information using the maximum entropy principle. To assess the quality of a distribution model, Mampaey et al. proposed to use the Bayesian Information Criterion (BIC) [19] that favors models with fewer parameters. Given a set of itemsets, one can compute the maximum entropy model and its BIC score. The goal is to find the set that best summarizes the data, i.e. with the smallest score. This approach is not relevant to the goal of our graph miner because MTV is looking for representative itemsets while we are searching for surprising patterns. However, having a set of representative patterns induces that patterns present in this set should not be surprising.

**Interesting features associations:** Bleco and Kotidis [20] proposed an entropy-based filter to locate interesting node features associations in a graph. The aim is to locate automatically the relationships of interest among the exponential number of combinations. Their work is based on the graph cube proposed by Zhao et al. [1]. Each aggregate network built from any combination of attributes is assigned an entropy value depicting the distribution of its nodes. However, the intuition behind this choice is not clearly stated. Then the authors use an entropy-based measure to navigate within the cube lattice. This measure is used to define whether an aggregation level must be extended.

### III. Preliminaries

#### A. Graph Cubes

First, we introduce formally the concept of multidimensional network, a graph with attributes on the nodes. This is the data structure on which the graph cube is based. Then in Section IV we formally define the patterns we are looking for in such a network.

**Definition 1:** A multidimensional network $N$ is a graph denoted as $N = (V, E, A)$, where $V$ is a set of vertices, $E \subseteq V \times V$ is a set of edges and $A = \{A_1, A_2, \ldots, A_n\}$ is a set of $n$ vertex-specific attributes, i.e. $\forall u \in V$, there is a multidimensional tuple $A(u) = (A_1(u), A_2(u), \ldots, A_n(u))$, where $A_i(u)$ is the value of the $i$-th attribute of the vertex $u$, with $1 \leq i \leq n$. $|A|$ is called the dimension of the network $N$ we consider.

The toy network was presented as an example in Section I in Figure 3a. An aggregate network or *cuboid* built from it represents a view from such a network where one or multiple attributes are ignored. This view contains the information about a specific combination of attributes. We first introduce the notions of equivalence between nodes and edges used to define how to merge them to form cuboids. Then we give the formal definition of an aggregate network as introduced by Zhao et al. [1].

**Definition 2:** Let $N = (V, E, A)$ be a multidimensional network and $A' = (A'_1, A'_2, \ldots, A'_n)$ a possible aggregation of $A$, where $A'_i = A_i$ or $\ast$. If $A'_i = \ast$ then the $i$-th dimension is ignored in the process. Let $u, v \in V$ be two nodes of the network that are candidates to be equivalent. We say that $u$ and $v$ are **equivalent** according to $A'$ if

$$\forall i \text{ such that } A'_i \neq \ast : A'_i(u) = A'_i(v).$$

Furthermore, we denote by $eq_{A'}(u, v)$ the function yielding a true value if $u$ and $v$ are equivalent according to $A'$, or a false value otherwise.

As an example let us consider the aggregation $A' = (\ast, \ast, \text{Profession})$ and the toy network with numbered nodes and edges in Figure 3. Then the nodes 2, 5 and 10 are equivalent because they denote the three lawyers of the network.

**Definition 3:** Let $N = (V, E, A)$ be a multidimensional network and $A' = (A'_1, A'_2, \ldots, A'_n)$ a possible aggregation of $A$. Let $e, f \in E$ be two edges of the network such that $e = (u_e, v_e), f = (u_f, v_f)$. We say that $e$ and $f$ are **equivalent** according to $A'$ if their end nodes are equivalent, i.e.

$$eq_{A'}(u_e, u_f) \land eq_{A'}(v_e, v_f) \lor eq_{A'}(u_e, v_f) \land eq_{A'}(u_f, v_e)$$

Similarly we denote by $eq_{A'}(e, f)$ the function yielding a true value if $e$ and $f$ are equivalent according to $A'$, or a false value otherwise.
Considering \( A' = (*,*,\text{Profession}) \) in our toy example, \( e_2 \) and \( e_5 \) are two equivalent edges with respect to \( A' \) as they both associate an engineer and a lawyer.

**Definition 4:** Let \( N = (V,E,A) \) be a multidimensional network and \( A' = (A_1', A_2', \ldots, A_n') \) a possible aggregation of \( A \), where \( A_i' = A_i \) or \( * \). Then the aggregate network with respect to \( A' \) is a weighted network \( N' = (V', E', A') \), where

1) To every equivalence set of nodes \( V_{eq} \) of \( V \), a node in the aggregate network \( v' \in V' \) is associated. The weight of \( v' \) is the number of equivalent nodes in \( V_{eq} \), i.e. \( w(v') = |V_{eq}| \). Therefore \( v' \) is called a condensed vertex.

2) To every equivalence set of edges \( E_{eq} \) of \( E \), an edge in the aggregate network \( e' \in E' \) is associated. The weight of \( e' \) is the number of equivalent edges in \( E_{eq} \), i.e. \( w(e') = |E_{eq}| \). Therefore \( e' \) is called a condensed edge.

From a multidimensional network multiple aggregations can be defined to obtain as many aggregate networks. We could wonder how these different networks are linked to each other. We introduce here the definition of a graph cube, based on [1].

**Definition 5:** Given a multidimensional network \( N = (V,E,A) \), the graph cube is obtained by restructuring \( N \) in all possible aggregations of \( A \). One corresponding aggregate network \( N' \) is associated to each aggregation \( A' \) of \( A \). An aggregation of a multidimensional network \( N = (V,E,A) \) is called a cuboid.

In the remainder of this paper we use the terms cuboid, graph and network to denote the multidimensional network defined by an aggregation. Figure 2 illustrates the concept of a graph cube lattice, that contains all the possible aggregate networks or cuboids. The numbers denote the number of nodes and edges in the networks. We say that a cuboid \( S' \) is an ancestor of another cuboid \( S \) if \( S \) is defined over all the attributes of \( S' \) and one or multiple other attributes. For example, \((*,*,\text{Profession})\) is an ancestor of \((*,\text{Location},\text{Profession})\). The most aggregate network represented at the top of the lattice is called the apex. It is the ancestor of all the cuboids. Furthermore, we call \( S' \) a direct ancestor of \( S \) if \( S \) is defined over all the attributes of \( S' \) and exactly one other attribute.

**B. Mapping Pattern Mining in Graphs to Itemset Mining**

In this work, we propose an approach that maps the problem of finding interesting aggregate edges in the graph cube to the problem of finding interesting itemsets in a transactional database. The idea is that each edge in the original graph can be mapped to a transaction and a pair of attribute values can be mapped to an item. Hence a transaction consisting of a set of items is equivalent to a set of pairs of attribute values defining an edge.
Tuples of transactions $$\{(\text{FF, NLBE, TL}), (\text{FF, NLBE, LE})\}$$ and $$\{(\text{FF, DE, T}), (\text{FF, DE, L})\}$$ are different.

Table I: Mapping from the toy network to its corresponding transactional database

<table>
<thead>
<tr>
<th>Edge</th>
<th>Tuples</th>
<th>Transactions</th>
</tr>
</thead>
<tbody>
<tr>
<td>$$e_1$$</td>
<td>(F, BE, T), (M, NL, T)</td>
<td>(FF, NLBE, TE), (FF, NLBE, LE)</td>
</tr>
<tr>
<td>$$e_2$$</td>
<td>(M, DE, E), (F, DE, T)</td>
<td>(M, DE, E), (F, DE, L)</td>
</tr>
</tbody>
</table>

Networks of the graph cube lattice but the apex. Figure 4a and Figure 4b present the networks associated with the cuboids $$S = (\text{Gender, *}, \text{Profession})$$ and $$S' = (\text{Gender, *, *})$$ respectively. Then the proportion of the tuple (Male, Lawyer) with respect to $$S'$$ is given by

$$p_{S,S'}([\text{M, Law}]) = \frac{W_T([\text{M}, \text{Law}])}{W_T([\text{M}])} = \frac{2}{5}$$

Please note that the support of a pattern in a cuboid is the weight of the pattern in such a network. For example, in Figure 4a the support of the pattern $$\{\text{(Female, Lawyer), (Female, Teacher)}\}$$ is 1 and in Figure 4b the support of $$\{(\text{Male}, \text{Female})\}$$ is 9. Both figures denote aggregate networks built from the toy example. In Section V we define the interestingness of a pattern based on its support.

V. SURPRISING PATTERNS DISCOVERY

To quantify how surprising a pattern is, we use an hypothesis testing framework. For a particular pattern $$P$$ we compute the probability to observe $$\text{supp}(P)$$ occurrences of it, given some prior knowledge we have on the data. This knowledge is based on the ancestors of the network where $$P$$ lies.

A. Pattern probability

First, we define the probability of a node having certain attributes defined by a tuple $$T$$ with respect to a more aggregated view of the network and denote it by $$p$$. The probability $$p_{S,S'}(T)$$ of a node having attribute values $$T$$ over $$S$$ is the fraction of nodes having attribute values $$T'$$ among the nodes having attribute values $$T'$$ where $$T'$$ is defined over $$S' \subset S$$.

$$p_{S,S'}(T) = \frac{\text{supp}_{S'}(T)}{\text{supp}_{S}(T)}$$

Example 3: Let us consider again our toy network. Figure 4a and Figure 4b present the networks associated with the cuboids $$S = (\text{Gender, *}, \text{Profession})$$ and $$S' = (\text{Gender, *, *})$$ respectively. Then the proportion of the tuple (Male, Lawyer) with respect to $$S'$$ is given by

$$p_{S,S'}([\text{M, Law}]) = \frac{W_T([\text{M}, \text{Law}])}{W_T([\text{M}])} = \frac{2}{5}$$
where \( W_V \) and \( W_{V'} \) are the functions mapping a tuple of attributes to its weight in the networks corresponding to the cuboids \( S \) and \( S' \) respectively.

A possible definition for the probability of a pattern is inspired by the partition models [21] and adapted to the graph-cube lattice we are working with. The idea in the itemset mining setting is to assume that all the elements of a partition of the items occur independently from each other. Here we consider an independence model, i.e. a particular case of a partition model.

In our setting, let \( P' = (T'_1, T'_2) \) be the pattern \( P \) restricted to \( S' \), i.e. \( P \) from which we remove the values of the attributes \( S \setminus S' \). The idea is to compute the probability of observing \( P \) given that we observe \( P' \) a certain number of times. We calculate this probability under the null hypothesis that the data is generated from an independence model. In this model we assume that each node has the same probability to be drawn. As a consequence, the probability of observing an edge \( P = (T_1, T_2) \) is equal to the product of the proportions of its two end points \( T_1 \) and \( T_2 \) with respect to \( S' \). If we take an edge at random in \( G \), the probability of the end points attribute values to match the pattern \( P \) given that we know the support of \( P' \) is given by Equation 1.

\[
\Pr(P \mid N') = p_{S,S'}(T_1)p_{S,S'}(T_2) \tag{1}
\]

This is the probability that one of the \( \text{supp}(P') \) edges is an edge between \( T_1 \) and \( T_2 \). \( P \) has a single equivalent edge in \( N' \). This is the edge whose nodes have the same properties as the nodes of \( P \) except the aggregated one(s). Several edges in \( G \) can have the same ancestor edge in \( N' \). Therefore we can define the probability to draw an edge \( P \) from its corresponding ancestor edge which has a support greater or equal to the support of \( P \).

**Example 4:** We want to compute the probability that an edge lies between a woman in Belgium and a man in the Netherlands, given that the edge lies between a man and a woman. Let us consider the aggregate edge \( P' = [(\text{Male}), (\text{Female})] \), which has a weight of 9 and is highlighted in Figure 4b and Figure 5b. It is distributed into 5 aggregate edges in the cuboid (Gender, Location, *). Each of these aggregate edges has its own probability to be drawn from the 9 original edges aggregated into one in the cuboid (Gender, *, *). The probability that the aggregate edge \( P_1 = [(\text{Female}, \text{BE}), (\text{Male}, \text{NL})] \) is drawn from the 9 edges is given by \( \frac{2}{9} \times \frac{2}{9} = 0.16 \).

Please note that \( \text{supp}(P_1) = 4 \) and that \( P_1 \) is as likely to be drawn as \( P_2 = [(\text{Female}, \text{BE}), (\text{Male}, \text{DE})] \) while \( \text{supp}(P_2) = 1 \).

However, this quantity is not always correct. Let us consider the (Gender, *, *) network in Figure 4b that contains a total of 10 nodes and 13 edges. Its only ancestor is the apex that contains a self-loop. Considering that we draw an edge from one of the 13 edges aggregated in the self-loop, the probability to get the edge \([(\text{Male}), (\text{Female})]\) is equal to the product of the proportions of (Male) and (Female), that is 0.25. The two other edges of this network have the same probability to be drawn, leading to a sum of probabilities equal to 0.75. The missing fourth comes from the \([(\text{Male}), (\text{Female})]\) edge. Indeed as we draw an edge by drawing its two end points successively we should add the probability to draw a (Female) node after a (Male) node. Our drawing process induces an order between the nodes and therefore a direction to the edges.

Generally speaking, this situation occurs when drawing a non self-loop edge \((T_1, T_2)\) whose equivalent edge in an ancestor cuboid is a self-loop. Hence we choose to define the probability of observing an undirected edge \((T_1, T_2)\) as the sum of the probability to draw \( T_1 \) then \( T_2 \) and the probability to draw \( T_2 \) then \( T_1 \). This means that in the toy example we mentioned above we should add the probability to draw a man then a woman and the probability to draw a woman then a man when drawing from the unique self-loop edge of the apex. Hence the probability to draw an edge between a man and a woman is equal to 0.5. This solution induces a bias towards non self-loops being more likely than self-loops.

### B. Pattern interest

The interest of a pattern can be defined in various ways. In this work, we say that a pattern is interesting if it is surprising with respect to some prior knowledge. We derive the probability that a pattern \( P \) has indeed a support equal to the observed support \( \text{supp}(P) \), given that \( P \) is drawn from the aggregate edge, in the ancestor \( N' \), equivalent to \( P \). We model the support of \( P \) as a random variable following a binomial distribution. The Bernoulli trial is the following: an edge is drawn from the aggregate edge in \( N' \) equivalent to \( P \) with probability \( p = \Pr(P \mid N') \) as given in Equation 1.

Equation 2 defines the probability to observe \( \text{supp}(P) \) occurrences of \( P \) in a network \( N \) defined over \( S \) with respect to an ancestor cuboid \( S' \subset S \).

![Figure 5: The networks corresponding to the cuboids (*, Location, *) and (Gender, Location, *) built from the toy network](image)
The probability of observing $supp(P)$ for $P = [(\text{Female, BE}), (\text{Male, NL})]$ given each of its direct ancestor networks in Example 5 is best explained by the $(\ast, \text{Location}, \ast)$ network probability. Hence we define the interestingness of a pattern to be equal to the highest probability computed from its ancestors. We define it as the quantity that best explains $supp(P)$ according to the independence model and denote it by $int(P)$. The lower the interestingness, the more surprising the pattern is.

$$int(P) = \max_{S' \subseteq S_{anc}} \Pr(supp(P))$$  

(3)

We calculated the probability of observing $supp(P)$ for $P = [(\text{Female, BE}), (\text{Male, NL})]$ given each of its direct ancestor networks in Example 5. This pattern is best explained by the $(\ast, \text{Location}, \ast)$ network with a probability of 0.23. Hence the interestingness of $P$ is equal to 0.23.

C. Search algorithm

Algorithm 1 performs an exhaustive search throughout every aggregate network of the multidimensional network $\mathcal{N}$ using a probability threshold $\theta$. Any aggregate edge having the probability of its support above this threshold is not interesting.

```
Algorithm 1: Search for surprising patterns in the graph cube lattice

Input : $\mathcal{L}$ the lattice containing all the aggregate networks built from a given network
Input : $\theta$ the aggregate edge probability threshold
Output: The set of surprising aggregate edges in any network of $\mathcal{L}$ with respect to $\theta$

1 patterns $\leftarrow \emptyset$
2 /* Most aggregated network is apex */
3 for $\mathcal{N} \leftarrow \mathcal{L} \setminus \text{apex}$ do
4   $(V, E, A) \leftarrow \mathcal{N}$
5   for $e \in E$ do
6     interesting $= \text{true}$
7     $p = 0$
8     ancestors $= \{(V', E', A') \mid A' \in A, |A'| = |A| - 1\}$
9     for $\mathcal{N}' \leftarrow \text{ancestors}$ do
10    $p = \max(p, \text{probability}(\mathcal{N}, \mathcal{N}', e))$
11    if $p \geq \theta$ then
12      interesting $= \text{false}$
13      break
14    end
15   if interesting then
16     patterns $\leftarrow \text{patterns} \cup (e, p)$
17   end
18 end
19 return patterns sorted by probability
```

VI. EXPERIMENTS

We test our measure of interestingness in attributed graphs on two different datasets. The first one is collected by the GroupLens research lab at the University of Minnesota\(^1\) and consists of a million of ratings given by MovieLens\(^2\) users to movies. Demographic information on the users is also provided. We use the dataset to build a network where the nodes correspond to users and we create an edge between two users if they have the same cinematographic tastes.

The second dataset is an enhancement of the 10 millions ratings MovieLens dataset where information about the movies from the Internet Movie Database\(^3\) (IMDb) and Rotten Tomatoes\(^4\) (RT) has been added. Therefore we build a network where a node represents a movie and there is

---

\(^1\)GroupLens Research group, https://www.grouplens.org
\(^2\)MovieLens, movie recommendations, https://www.movielens.org
\(^3\)Internet Movie Database, http://www.imdb.com
\(^4\)Rotten Tomatoes - movie critic reviews, https://www.rottentomatoes.com
an edge between two movies if they are liked by a certain number of same users. We present in this section how we construct these two networks and the experiments we conducted on them.

A. Network construction

The MovieLens dataset: The MovieLens1M dataset encompasses some demographic information about the users, namely their age, gender, occupation and location. The dataset consists of a million of ratings given between 2000 and 2003, its characteristics are given in Table II. The rating scale is an integer from 1 to 5 stars. We include users that gave at least 20 ratings. From this dataset we build a multidimensional network \( \mathcal{N}_1 = (V_1, E_1, A_1) \). The set of nodes \( V_1 \) is directly the set of provided users. The attributes of the nodes \( A_1 \) are the user features provided in the dataset. The age of a user belongs to one of the 7 age groups while there are 21 possible occupations. The location is given as a zip-code and we categorized it into U.S. states. The set of edges \( E_1 \) requires a similarity measure between users to be built.

We choose to put an edge between two users if they like a certain number of movies in common. We arbitrarily say that a user likes a movie if they rate it 4 or above. We define the similarity between two users to be the number of movies they both rated 4 stars or more. However, one can expect from a movie dataset that the function depicting the number of high ratings per movie follows a power law, a few blockbusters having a lot of high ratings while the rest of the movies is highly less rated at all. Figure 6 presents such a function. As expected the curve follows a power law and we can observe the long tail of the distribution.

Since the blockbusters will favor a higher similarity between all the users we choose to remove them. The function presented in Figure 6 shows an abrupt slope near the 500 users. Therefore we consider as blockbuster in this dataset the movies having at least 500 ratings of 4 stars or more and remove them. We refer to the obtained dataset as the pruned MovieLens1M dataset. Then we need to define a threshold from which the similarity measure yields an edge in the network. For this purpose we show in Figure 7 the relation between a similarity and the number of pairs of users having this similarity. We removed similarities 1 to 5 as they have a very high amount of pairs of users in order to observe the curve in more details.

We observe that this relation also follows a power law, a high number of pairs having a low similarity while a few pairs have a high similarity. We also observe that the slope becomes abrupt around a similarity of 20. Therefore in order to reduce the number of edges in the newly created network we set the similarity threshold to 20. The statistics about the obtained network \( \mathcal{N}_1 \) are given in Table II.

The extended MovieLens dataset: The second International Workshop on Information Heterogeneity and Fusion in Recommender Systems [22] (HetRec 2011) released several datasets among which the extended MovieLens dataset. It consists of the MovieLens10M dataset enhanced by more information about the movies from IMDb and RT such as the country of origin and the different user and critics ratings. Statistics about this dataset are presented in Table III The ratings were given from 1995 to 2009 on the MovieLens website and range from 0.5 to 5 stars.

For this network \( \mathcal{N}_2 = (V_2, E_2, A_2) \), we model the nodes \( V_2 \) to be the movies. Each movie has three attributes: its year of release, its country of origin and the average top critic. The original average RT top critic is given on a scale from 0 to 10 but we categorize it into 5 classes: very bad, bad, average, good, very good. We also categorize the year of release by replacing them by their respective decade. Finally...
we have $A_2 = (\text{critic, country, decade}).$ As for the MovieLens1M dataset, we need a similarity measure to build the set of edges $E_2.$

Similarly to the approach described above, an edge is put between two movies if they have a certain number of users that gave them a rating of 4 stars or more in common. The function depicting the number of movies rated 4 or more per user also follows a power law. A few users took the time to rate a high number of movies or like a lot of them while most users express their preferences for a small number of movies

Therefore a few users are easily similar to a lot of other users by liking a high number of movies. We decide to remove them by not taking into account users that gave a high rating to more than 400 movies. Then the similarity threshold is empirically defined in a similar manner as we did for the MovieLens1M dataset. We set the similarity threshold to 40. The statistics about the obtained network $N_2$ are given in Table III.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Users</th>
<th>Movies</th>
<th>Movie features</th>
<th>Ratings</th>
</tr>
</thead>
<tbody>
<tr>
<td>MovieLens10M</td>
<td>2100</td>
<td>10,200</td>
<td>(critic, country, decade)</td>
<td>860,000</td>
</tr>
<tr>
<td>MovieLens1M</td>
<td>678</td>
<td>318</td>
<td>similarities</td>
<td>10985</td>
</tr>
</tbody>
</table>

Table III: Statistics about the extended MovieLens10M dataset and network $N_2$ built from it

<table>
<thead>
<tr>
<th>Cuboid</th>
<th>Pattern</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Gender)</td>
<td>(M), (M)</td>
<td>0</td>
</tr>
<tr>
<td>(Gender)</td>
<td>(M), (F)</td>
<td>$2.31 \times 10^{-292}$</td>
</tr>
<tr>
<td>(Age)</td>
<td>[35], (18)</td>
<td>$1.4 \times 10^{-71}$</td>
</tr>
<tr>
<td>(State)</td>
<td>[GU], (CA)</td>
<td>$3.88 \times 10^{-40}$</td>
</tr>
<tr>
<td>(Gender)</td>
<td>[F], (F)</td>
<td>$2.1 \times 10^{-60}$</td>
</tr>
<tr>
<td>(Age, State)</td>
<td>[50], (PA), [25], (CA)</td>
<td>$1.95 \times 10^{-4}$</td>
</tr>
<tr>
<td>(Age)</td>
<td>[18], (50)</td>
<td>$3.12 \times 10^{-7}$</td>
</tr>
<tr>
<td>(State)</td>
<td>[ICA], (LA)</td>
<td>$1.75 \times 10^{-34}$</td>
</tr>
<tr>
<td>(Age)</td>
<td>[18], (56)</td>
<td>$1.09 \times 10^{-49}$</td>
</tr>
<tr>
<td>(Age)</td>
<td>[25], (25)</td>
<td>$9.42 \times 10^{-49}$</td>
</tr>
</tbody>
</table>

Table IV: Top 10 most interesting patterns in the MovieLens1M dataset reported by our miner

In this section we present the patterns output by our miner on the two MovieLens datasets. We interpret the results of our miner on both datasets and we show the limitations of the MINI algorithm.

Regarding the MovieLens1M dataset where nodes are users, the 10 most interesting edges according to the probability measure are given in Table IV. We can observe that the three aggregate edges of the gender cuboid are output. It seems to be the case that attributes having a small set of values are more likely to be surprising.

As for the extended MovieLens10M dataset where nodes are movies, the 10 most interesting edges are presented in Table V. We can observe that 4 of the edges are between top critic values. It seems to confirm the fact that attributes with a small set of values are more likely to be output as surprising. A possible interpretation of these results is the following. As the second most interesting edge lies between very bad and good movies according to the average top critic ratings, we can infer that people rating the movies in this dataset do not agree with the top critics.

We also ran the MINI algorithm on the two datasets. Results are given in Table VI and Table VII. As the algorithm uses a greedy heuristic to add the itemsets to the set of interesting patterns, we set the maximum number of iterations to 10,000. On both datasets the results are not really interesting, MINI reports the relations that often occur and that have a relatively high p-value. Moreover MINI did not return 10 patterns. Better results can be obtained by increasing the maximum number of iterations. But we report here only these results to illustrate the fact that MINI is greedy and hence can be fooled by unsurprising but frequent patterns.

<table>
<thead>
<tr>
<th>Cuboid</th>
<th>Pattern</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Gender)</td>
<td>(M), (M)</td>
<td>0.34</td>
</tr>
<tr>
<td>(Gender)</td>
<td>(F), (F)</td>
<td>0.34</td>
</tr>
<tr>
<td>(Age)</td>
<td>[25], (25)</td>
<td>0.74</td>
</tr>
<tr>
<td>(Age)</td>
<td>[35], (35)</td>
<td>0.74</td>
</tr>
</tbody>
</table>

Table VI: Top 10 most interesting patterns in the MovieLens1M dataset reported by MINI

<table>
<thead>
<tr>
<th>Cuboid</th>
<th>Pattern</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Critic)</td>
<td>[USA], (USA)</td>
<td>8.85 $\times 10^{-11}$</td>
</tr>
<tr>
<td>(Critic)</td>
<td>[USA], (Good)</td>
<td>0.74</td>
</tr>
<tr>
<td>(Critic)</td>
<td>[Good], (Very good)</td>
<td>0.74</td>
</tr>
<tr>
<td>(Decade)</td>
<td>[2000], (2000)</td>
<td>0.74</td>
</tr>
<tr>
<td>(Decade)</td>
<td>[1990], (1990)</td>
<td>0.74</td>
</tr>
</tbody>
</table>

Table VII: Top 10 most interesting patterns in the extended MovieLens10M dataset reported by MINI

VII. CONCLUSION

Our objective was to study the patterns that can be found in attributed graphs. We based our work on the graph cube...
data model and proposed a hypothesis testing framework to evaluate how surprising the found patterns are with respect to an independence model. We showed the relationship between our pattern mining framework and the frequent itemset mining literature. Moreover we proposed a mapping from attributed graphs to transactional databases. We compared the frequent itemset mining algorithm MINI on transactional databases with our method on attributed graphs theoretically. Furthermore, we gave the interpretability of the results obtained on two MovieLens datasets with MINI and with our method.

Many opportunities for extending this work exist. Experiments on synthetic datasets and other real datasets can be conducted to understand in more depth the patterns that can be found in the graph cube lattice. The maximum entropy model on binary databases could be adapted to attributed graph data to provide a generic framework for mining patterns in networks. The notion of pattern can be extended to heterogeneous features associations. For instance a pattern could consist of the number of relationships between Belgians and engineers.

REFERENCES


