Mining Association Rules in Graphs based on Frequent Cohesive Itemsets

Tayena Hendrickx, Boris Cule, Pieter Meysman,
Stefan Naulaerts, Kris Laukens, and Bart Goethals

University of Antwerp, Belguim,
firstname.lastname@uantwerp.be

Abstract. Searching for patterns in graphs is an active field of data mining. In this context, most work has gone into discovering subgraph patterns, where the task is to find strictly defined frequently re-occurring structures, i.e., node labels always interconnected in the same way. Recently, efforts have been made to relax these strict demands, and to simply look for node labels that frequently occur near each other. In this setting, we propose to mine association rules between such node labels, thus discovering additional information about correlations and interactions between node labels. We present an algorithm that discovers rules that allow us to claim that if a set of labels is encountered in a graph, there is a high probability that some other set of labels can be found nearby. Experiments confirm that our algorithm efficiently finds valuable rules that existing methods fail to discover.

1 Introduction

Discovering interesting patterns in graphs is a popular data mining task, with wide applications in social network analysis, bioinformatics, etc. In traditional approaches, the dataset consists of a (very large) single graph and the task is to find frequent subgraphs, i.e., reoccuring stuctures consisting of labelled nodes frequently interconnected in exactly the same way. However, the concept of frequent subgraphs is not flexible enough to capture all patterns. Another problem with the subgraph mining approaches is that they are also typically computationally complex since they are forced to deal with the graph isomorphism problem. For small graphs, isomorphism checking is still manageable, but for large graphs such checks become computationally very expensive.

In this work, we therefore base our study on the Frequent Cohesive Itemset (*FCI*) approach proposed by Hendrickx et al. [6]. A frequent cohesive itemset is defined as a set of node labels that are all, as individual items, frequent, and are, on average, tightly connected to each other, but not necessarily always connected in exactly the same way. In this paper, we introduce the concept of association rules into this setting. More formally, the aim is to generate rules of the form *if* X *occurs,* Y *occurs nearby*, where $X \cap Y = \emptyset$ and $X \cup Y$ is a frequent cohesive itemset. Such rules provide the end user with much more information about correlations and interactions between node labels than itemsets alone. Consider,

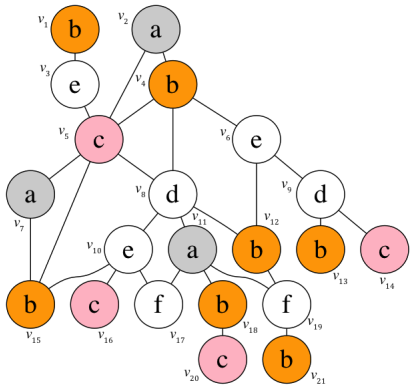


Fig. 1: A toy graph dataset.

for example, the graph given in Figure 1, and suppose ab is a frequent cohesive itemset (for simplicity, we denote an itemset $\{a, b\}$ as ab in our examples). Note that this pattern will not necessarily be discovered by subgraph mining, as nodes labelled a and b are not always connected by an edge. Based on this itemset, we can generate rule *if a node labelled 'a' occurs, a node labelled 'b' occurs nearby*, which is, as we can see, true for every node labelled a. However, we can also generate rule *if a node labelled 'b' occurs, a node labelled 'a' occurs nearby*, which does not necessarily hold, since some nodes labelled b are relatively far from the nearest node labelled a. If we only mine frequent cohesive itemsets, this type of information would remain undiscovered.

Given an association rule $X \Rightarrow Y$, we define its confidence as the average distance from an occurrence of X to the nearest occurrence of Y. Inspired by a similar approach to mining association rules in sequential data by Cule and Goethals [4], we develop an efficient algorithm to mine association rules based on frequent cohesive itemsets. The main two features of the algorithm are that it allows us to mine itemsets and association rules in parallel (rather than first finding all itemsets, and then all association rules, as is the case in most traditional approaches), and that we only need to compute the confidence of a simple class of rules, which can then be used to quickly evaluate all other rules. Experiments show that our algorithm finds interesting rules within acceptable runtime.

2 Related Work

Mining knowledge from graph structured data is an active research topic in data mining. A survey of the early graph-based data mining methods is given by Washio and Motada [16]. The traditional way of finding patterns in graphs is limited to searching for frequent subgraphs, i.e., reoccuring patterns within which nodes with certain labels are frequently interconnected in exactly the same way. The first attempts to find frequent subgraphs in a single graph were made

by Cook and Holder [3]. Meanwhile, Motoda and Indurkhya [18] developed an approach for a setting where the dataset consists of many (typically smaller) graphs, rather than a single large one. Further attempts at subgraph mining have been made by Yan and Han [17], Nijssen and Kok [14], and Dehaspe and Toivonen [5]. Although algorithms for discovering frequent subgraphs find patterns that are useful in many applications, they suffer from two main drawbacks — costly isomorphism testing and an enormous number of generated candidates since both edges and nodes must be added to the pattern.

To tackle these problems, Hendrickx et al. [6] recently proposed the *FCI* algorithm for mining frequent cohesive itemsets in graphs. An itemset is defined as a set of node labels which often occur in the graph and are, on average, tightly connected to each other. By searching for itemsets rather than subgraphs, the costly isomorphism tests are avoided, and the number of candidate patterns is reduced. Khan et al. [12] proposed another itemset mining approach for graphs, where node labels are, according to given probabilities, propagated to neighbouring nodes. Labels are thus aggregated, and can be mined as itemsets in the resulting graph.

The concept of association rule mining was first introduced by Agrawal et al. [2]. An association rule, denoted $X \Rightarrow Y$, expresses an observation that an occurrence of pattern X implies an occurrence of pattern Y. In the setting of transaction databases, the strength of a rule was measured in terms of its confidence which determines how frequently itemset Y appears in transactions that contain X. Since then, association rules have been applied in a variety of settings, with various definitions of the confidence measure. Cule and Goethals [4] successfully applied association rule mining on sequential data, where they defined the confidence of a rule in terms of how far on average from an occurrence of itemset X one has to look in order to find itemset Y.

In this paper, however, we apply the concept of association rules to graph data. In this setting, Inokuchi et al. [9] proposed an Apriori-based algorithm, called *AGM*, for mining frequenty appearing induced subgraphs in a given graph dataset and association rules among such subgraphs. In contrast with our work, they mine rules of the form $G_b \Rightarrow G_h$ where G_b and G_h are subgraphs, rather than itemsets. In another work, Inokuchi et al. [10] proposed an approach to mine frequent graph structures and the association rules among them embedded in massive graph structured transaction data. Here, the graph data is transformed in such a way that traditional association rule mining techniques can be applied, but the resulting rules still contain subgraph patterns.

3 Problem Setting

In this section, we define the concept of association rules in graph data based on frequent cohesive itemsets. For simplicity of presentation, we assume that the graph consists of labelled nodes and unlabelled egdes, and we focus on connected graphs with at most one label per node. However, we can also handle input graphs that contain nodes with multiple labels, by transforming each such node

into multiple nodes each carrying one label and connecting all of them to a central dummy node.

As our work is based on the Frequent Cohesive Itemset mining approach of Hendrickx et al. [6], we start with a reproduction of some of the necessary definitions and notations. We define a graph G as a set of nodes $V(G)$ and a set of edges $E(G)$. Each node $v \in V(G)$ carries a label $l(v) \in S$, where S is the set of all labels. For a label $i \in S$, we denote the set of nodes in the graph carrying this label as $L(i) = \{v \in V(G) | l(v) = i\}$. We define the frequency of a label $i \in S$ as the probability of encountering that label in G, or $freq(i) = \frac{|L(i)|}{|V(G)|}$. From now on, we will refer to labels as items, and sets of labels as itemsets.

For an itemset X, the set of all nodes labelled by an item in X is denoted by $N(X) = \{v \in V(G) | l(v) \in X\}$. In order to compute the cohesion of an itemset X we must look, for each occurence of an item in X, for the nearest occurence of all other items in X. For a node v, we define the sum of all these smallest distances as $W(v, X) = \sum_{x \in X} min_{w \in N(\{x\})} d(v, w)$, where $d(v, w)$ is the length of the shortest path from node v to node w. The average of such sums for all occurences of items making up itemset X is expressed as

$$\overline{W}(X) = \frac{\sum_{v \in N(X)} W(v, X)}{|N(X)|}.$$

The cohesion of an itemset X, where $|X| \geq 2$, is defined as the ratio of the itemset size and the average sum of the smallest distances defined above:

$$C(X) = \frac{|X| - 1}{\overline{W}(X)}.$$

Note that the itemset size is reduced by one, since each considered node already carries one item in X. If $|X| < 2$, we define $C(X)$ to be equal to 1.

Cohesion measures how close to each other the items making up itemset X are on average. If the items are always directly interconnected by an edge, the sum of these distances for each occurence of an item in X will be equal to $|X| - 1$, as will the average of such sums, and the cohesion of X will be equal to 1.

Finally, an itemset X is considered *frequent cohesive* if, given user defined thresholds for frequency (*min_freq*) and cohesion (*min_coh*), it holds that $\forall x \in X : freq(x) \geq min\_freq$ and $C(X) \geq min\_coh$. Two optional parameters, *minsize* and *maxsize*, can be used to limit the size of the discovered itemsets.

In this setting, our goal is to generate rules of the form *if X occurs, Y occurs nearby*, where $X \cap Y = \emptyset$ and $X \cup Y$ is a frequent cohesive itemset. In the rest of this paper, we will denote a rule in the traditional way as $X \Rightarrow Y$, with X the body of the rule and Y the head of the rule. It is clear that the closer the items of Y occur to the items of X in the graph, the higher the confidence value of the rule should be. More formally, in order to compute the confidence of the rule, we must compute the average sum of minimal distances for $X \cup Y$, but only from the point of view of items making up itemset X, i.e.,

$$\overline{W}(X, Y) = \frac{\sum_{v \in N(X)} W(v, X \cup Y)}{|N(X)|}.$$

The confidence of a rule can then be defined as

$$c(X \Rightarrow Y) = \frac{|X \cup Y|-1}{\overline{W}(X,Y)}. \quad (1)$$

Given a confidence threshold *min_conf*, we consider rule $X \Rightarrow Y$ confident if $c(X \Rightarrow Y) \geq$ *min_conf*.

Let us now apply the definitions introduced above on the graph depicted in Figure 1. Assume that *min_freq*, *min_coh* and *min_conf* are set to 0.1, 0.5 and 0.6, respectively. Looking at itemset ab we first note that $freq(a) = \frac{3}{21} \approx 0.14$ and $freq(b) = \frac{7}{21} \approx 0.33$, which shows us that both a and b are frequent. Now let us have a look at the cohesion of itemset ab. According to our defintions, $|N(ab)|=10$. To compute the cohesion, we now search the neighbourhood of each node in $N(ab)$ which gives us $W(v_2, ab) = W(v_4, ab) = W(v_7, ab) = W(v_{11}, ab) = W(v_{15}, ab) = W(v_{18}, ab) = 1$, $W(v_{12}, ab) = W(v_{21}, ab) = 2$, $W(v_1, ab) = 3$ and $W(v_{13}, ab) = 4$. Subsequently, computing the average of the above minimal distances, we get $\overline{W}(ab) = \frac{17}{10} = 1.7$. We can now compute the cohesion of ab as $C(ab) = \frac{2-1}{1.7} \approx 0.58$, and conclude that itemset ab is sufficiently cohesive.

If we look at the occurences of a and b in the graph, we see that every node labelled a has a node labelled b nearby, but not vice versa. Therefore, the confidence of rule $a \Rightarrow b$ should be higher than that of rule $b \Rightarrow a$. According to our definitions, we see that for each node v labelled a, $W(v, ab) = 1$. Therefore, $\overline{W}(a, b) = \frac{3}{3} = 1$, and $c(a \Rightarrow b) = 1$ which means that rule $a \Rightarrow b$ has, as expected, a confidence of 100%. Meanwhile, the sum of distances from each b to the nearest a is $\sum_{v \in N(\{b\})} W(v, ab) = 14$. It follows that $\overline{W}(b, a) = \frac{14}{7} = 2$ and $c(b \Rightarrow a) = 0.5$. We see that $a \Rightarrow b$ is a confident association rule, while $b \Rightarrow a$ is not, and we conclude that while an occurence of a b does not imply finding an a nearby, when we find an a, however, we can expect to find a b nearby.

4 Algorithm

In this section we present a detailed description of our algorithm for discovering confident association rules based on frequent cohesive itemsets. Unlike the traditional approaches, which need all frequent itemsets to be found before the generation process of the association rules can begin, we will generate the rules in parallel with the frequent cohesive itemsets.

We begin by giving a quick overview of the FCI algorithm [6], which will serve as a basis for our algorithm. The FCI algorithm, given in Algorithm 1, discovers frequent cohesive itemsets. Candidate itemsets are generated by applying depth-first search, using recursive enumeration. During this process, a candidate consists of two elements — items that make up the candidate, and all frequent items which still have to be enumerated, denoted as X and Y, respectively. The algorithm uses the *UBC* pruning function, which computes an upper bound on the cohesion of all candidates yet to be generated in a given branch of the search tree, and decides whether to proceed deeper into the search tree, or to prune the complete branch. When a frequent cohesive itemset is found in

Algorithm 1 FCI($\langle X, Y \rangle$) finds frequent cohesive itemsets

```
1: if UBC(⟨X, Y⟩) ≥ min_coh then
2:    if Y = ∅ then
3:       if X ≠ ∅ and |X|≥ minsize then GENERATE_RULES(X)
4:    else
5:       Choose a in Y
6:       if |X ∪ {a}|≤ maxsize then FCI(⟨X ∪ {a}, Y \ {a}⟩)
7:       if |X ∪ (Y \ {a})|≥ minsize then FCI(⟨X, Y \ {a}⟩)
8:    end if
9: end if
```

line 3, rather than outputting it, we proceed to mine association rules that can be generated from this itemset, as discussed below.

The most computationally costly step of the FCI algorithm is the computation of $\sum_{v \in N(X)} W(v, X)$, needed to evaluate the cohesion of itemset X. Computing this at each node in the search tree would be unfeasible, but it has been shown that such a sum for an itemset X can be expressed as a sum of separate sums of such distances for each pair of items individually [6]. FCI stores these sums between individual items in a matrix which is generated in the beginning of the algorithm. Since we are only interested in itemsets consisting of frequent items, the matrix will only contain the minimal distances between each pair of frequent items. Therefore, the matrix that will be generated will only contain $|F| \times |F|$ sums of smallest distances, where F are the frequent items.

On top of being used in the generation of the frequent cohesive itemsets, this matrix can help us easily compute the confidence of association rules $X \Rightarrow Y$. More formally, for each $x \in X$, we can find the sum of smallest distances to each $y \in Y$ seperately. This would allow us to efficiently compute $\overline{W}(X, Y)$, and thus $c(X \Rightarrow Y)$. However, we can optimise this process further, taking advantage of the fact that it is sufficient to limit our computations to rules of the form $x \Rightarrow X \setminus \{x\}$ (i.e., rules where the body consists of a single item), with $x \in X$, which can be generated from an itemset X. To compute the confidence of all other rules, we first note that

$$\sum_{v \in N(X)} W(v, X \cup Y) = \sum_{x \in X} \Big(\sum_{v \in N(\{x\})} W(v, X \cup Y) \Big).$$

The last part of the above expression can be rewritten as

$$\sum_{v \in N(\{x\})} W(v, X \cup Y) = \overline{W}(x, X \cup Y \setminus \{x\}) |N(\{x\})|,$$

which allows us to reformulate the formula for the average sum of minimal distances as

$$\overline{W}(X, Y) = \frac{\sum_{x \in X} \Big(\overline{W}(x, X \cup Y \setminus \{x\}) |N(\{x\})| \Big)}{|N(X)|}.$$

This, in turn, implies that

$$c(X \Rightarrow Y) = \frac{(|X \cup Y|-1)|N(X)|}{\sum_{x \in X} \left(\overline{W}(x, X \cup Y \setminus \{x\})|N(\{x\})| \right)}.$$

For the case where X contains just one element, i.e., $X = \{x\}$, this gives us

$$c(x \Rightarrow Y \cup X \setminus \{x\}) = \frac{|X \cup Y|-1}{\overline{W}(x, Y \cup X \setminus \{x\})},$$

and it follows that

$$c(X \Rightarrow Y) = \frac{|N(X)|}{\sum_{x \in X} \frac{|N(\{x\})|}{c(x \Rightarrow X \cup Y \setminus \{x\})}}. \tag{2}$$

As a result, once we have computed the confidence of all the rules of the form $x \Rightarrow X \setminus \{x\}$, with $x \in X$, we can evaluate all other rules $Y \Rightarrow X \setminus Y$, with $Y \subset X$, without looking at either the dataset or the distance matrix.

Let us now return to our example graph shown in Figure 1 and compute the confidence of rule $ab \Rightarrow c$, where abc is a frequent cohesive itemset. First we will compute the confidence of rules $a \Rightarrow bc$ and $b \Rightarrow ac$. We find that $\sum_{v \in N(\{a\})} W(v, abc) = 7$ and $\sum_{v \in N(\{b\})} W(v, abc) = 27$. Consequently, $c(a \Rightarrow bc) = \frac{3-1}{\frac{7}{3}} \approx 0.85$ and $c(b \Rightarrow ac) = \frac{3-1}{\frac{27}{7}} \approx 0.51$. Following Equation 2, we can compute $c(ab \Rightarrow c)$ as $\frac{10}{\frac{3}{0.85} + \frac{7}{0.51}} \approx 0.58$. If we use Equation 1 defined in section 3 for computing the confidence of the same rule, we get $c(ab \Rightarrow c) = \frac{3-1}{\frac{34}{10}} \approx 0.58$, which shows us that Equations 1 and 2 are equivalent. For reasons explained above, we will use Equation 2 in our algorithm.

The algorithm for generating the confident association rules is given in Algorithm 2. Having found a frequent cohesive itemset X, we first generate rules of the form $x \Rightarrow X \setminus \{x\}$, store their confidence values in memory and send all confident rules to the output (lines 1 - 4). We then generate all other rules, compute their confidence based on the stored confidences computed during the first stage, and output all confident rules (lines 5 - 7).

Algorithm 2 GENERATE_RULES(X) generates rules based on itemset X

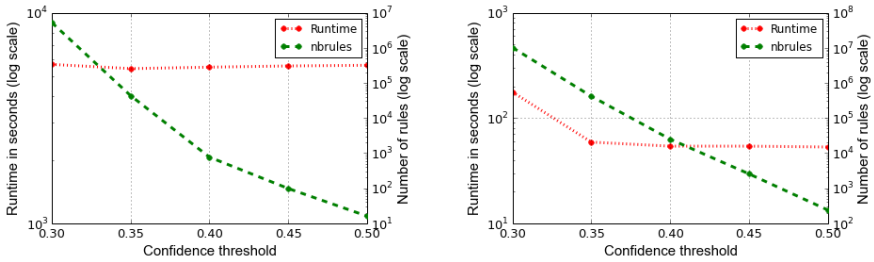
1: **for all** $x \in X$ **do**
2: **compute and store** $c(x \Rightarrow X \setminus \{x\})$
3: **if** $c(x \Rightarrow X \setminus \{x\}) \geq$ *min_conf* **then output** $x \Rightarrow X \setminus \{x\}$
4: **end for**
5: **for all** $Y \subset X$ with $|Y| \geq 2$ **do**
6: **if** $c(Y \Rightarrow X \setminus Y) \geq$ *min_conf* **then output** $Y \Rightarrow X \setminus Y$
7: **end for**

5 Experiments

In this section, we present the experimental results of our association rule mining algorithm. For our experiments, we used two different biological graph datasets. The first dataset concerns the human interactome. In this graph, each node is a protein and each edge is a direct interaction between two proteins. This network was obtained by fusing the BioGRID [15] database with the IntAct [11] and MINT [13] networks. Next, the protein nodes were labelled with the annotations contained in InterPro [7], which concerns common domains, protein families and active sites that are present in the corresponding protein. Note that each protein can contain multiple domains and can thus have more than one annotation label. The second dataset concerns the transcriptional regulatory network of Saccharomyces cerevisiae, commonly known as baker's yeast, which serves as an important model organism in life sciences. The yeast regulatory network was obtained from the YEASTRACT [1] database. Within this graph, each node is a yeast gene, and each edge corresponds to a regulatory interaction between a transcription factor gene and a target gene. Each node was labelled with gene ontology annotation information with regards to the biological process, molecular function and cellular location as obtained from UniProtKB [8]. As a result, here, too, each node can have multiple labels.

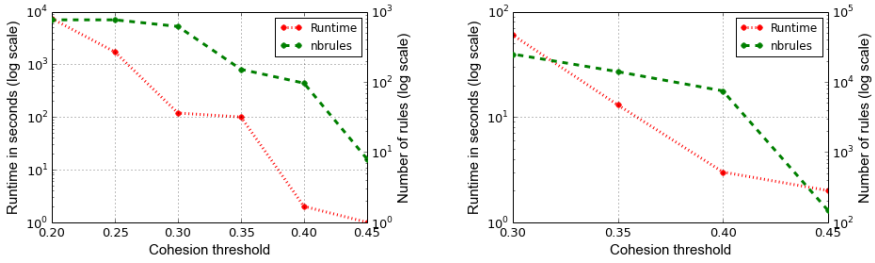
Since both datasets consist of nodes that can carry multiple labels, we first had to transform the given graphs. More formally, we expanded the given graph structure by replacing each node with a so-called dummy node carrying a unique label, and then connecting this node to a set of new nodes, each carrying one of the original labels. Note that due to the nature of this reconstruction, any two labels in the resulting graph will be at a distance of at least 2 from each other, as there will always be a dummy node in between. As a result, the maximal cohesion for any itemset in such a graph will be 0.5, and the same holds for the maximal confidence for any possible association rule. For the human interactome graph dataset, the transformed graph consisted of 64 090 nodes and 141 828 edges. The yeast regulatory network was transformed into a graph consisting of 21 314 nodes and 62 991 edges.

For all the experiments discussed below, we chose a frequency threshold that was low enough to guarantee a large number of frequent items, since we were only interested in the performance of our association rule miner. For this reason, we do not include the time needed to set up the distance matrix in our analysis, since this only needs to be done once, after which the matrix can be reused for various cohesion and confidence thresholds. The frequency threshold, *minsize* and *maxsize* were set to 0.002, 2 and 7, respectively, for the human interactome network, and to 0.001, 2 and 5, respectively, for the yeast network. In our first set of experiments, we also fixed the cohesion threshold in order to evaluate the effect of varying the confidence threshold. For the human interactome network we used $min\_coh = 0.2$, and for the yeast network $min\_coh = 0.3$. Figure 2 shows, for both datasets, the number of discovered rules and the runtimes needed to generate these rules for varying *min_conf* values. As expected, the number of discovered rules grows as we lower the confidence threshold, but the runtimes



(a) Human Interactome dataset. (b) Yeast dataset.

Fig. 2: Runtime and output size for varying confidence thresholds.



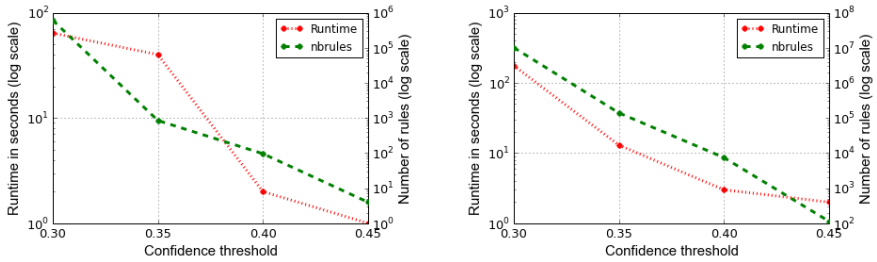
(a) Human Interactome dataset. (b) Yeast dataset.

Fig. 3: Runtime and output size for varying cohesion thresholds.

stay stable, since most time is spent mining itemsets and generating candidate rules, regardless of how many rules are actually found to be confident.

In the second set of experiments, we only vary the cohesion threshold, while we set the confidence threshold to 0.4 for both datasets. The results are shown in Figure 3 and confirm that varying the cohesion threshold does have an effect on runtimes. Naturally, the number of rules decreases as the cohesion threshold increases, since the number of frequent cohesive itemsets on which we base the rules also decreases.

Note that the cohesion of an itemset can be interpreted as a weighted average of the confidences of all association rules that can be generated from that itemset. In our third set of experiments, we varied both the cohesion and confidence thresholds, by setting $min\_conf = min\_coh$, thus finding those rules that had an above average confidence within the set of rules originating from the same itemset. The results are shown in Figure 4 and are similar to those given in Figure 3. Naturally, the number of rules now decreases faster, as the confidence threshold is raised together with the cohesion threshold. Finally, we note that in all experiments we managed to find a large number of confident association rules quickly and efficiently.



(a) Human Interactome dataset.

(b) Yeast dataset.

Fig. 4: Runtime and output size for varying cohesion and confidence thresholds, where *min_conf*=*min_coh*.

Let us now have a closer look at some of the discovered rules. Having shown the output to domain experts, it appeared that, for the human protein-protein interaction network, the majority of the discovered rules concern protein kinases. 99.4% of the rules included an annotation with the term 'kinase'. Kinase proteins are enzymes that can modify other proteins by phosphorylating specific residues on the target protein. Phosphorylation results in the addition of a phosphate group to the target protein. This is a common mechanism for many signal transduction pathways in all living cells, and thus a critical component of the protein-protein interaction network subjected to analysis here. For example, with *min_coh* and *min_conf* both set to 0.3 one example rule, existing of items which occur often in data but which form a non-trivial set, is *Tyrosine-protein kinase, catalytic domain* ⇒ *Protein kinase, ATP binding site, Serine-threonine/tyrosine-protein kinase catalytic domain, SH2 domain* with a confidence of 0.42. Proteins that contain a catalytic domain for tyrosine-protein kinase activity will phosphorylate a tyrosine residue on their target. The phosphorylation reaction typically requires ATP, and thus kinases will often co-occur with ATP-binding sites within the network. The annotation of 'Serine-threonine/tyrosine-protein kinase catalytic domain' is a higher level one that contains all instances of 'Tyrosine-protein kinase, catalytic domain' and thus is redundant with this annotation. The presence of the 'SH2 domain' in the head of the association rule is much more interesting. While the SH2 domain is not directly related to protein kinase activity itself, it allows a protein to dock phosphorylated tyrosine residues.

In the yeast dataset, with *min_coh* and *min_conf* set to 0.3, we found rule *heterocycle catabolic process* ⇒ *cellular nitrogen compound catabolic process*, with a confidence of 0.5. Genes annotated with 'heterocycle catabolic process' code for proteins that catalyse reactions for breaking down heterocyclic compounds. Heterocyclic molecules contain ring structures that consist of at least two types of atoms. The high confidence of this rule suggests that all genes involved in the degradation of heterocycles are also associated with 'cellular nitrogen compound catabolic process' in our network. This is to be expected as almost all heterocycles in living organisms consist of carbon and nitrogen atoms. Thus any gene

involved in the breakdown of heterocycles will also be associated in the breakdown of compounds that include carbon and nitrogen. Another logical rule is *sulfur compound biosynthetic process* $\Rightarrow$ *cellular nitrogen compound biosynthetic*, with a confidence of 0.48. This association rule suggests that there is a direct link between genes involved in the synthesis of compounds containing sulfur and genes involved in the synthesis of nitrogen compounds. This can be explained by the fact that the primary carriers of sulfur in yeast are the amino acids methionine and cysteine, both of which also contain nitrogen. We conclude that our algorithm discovers both expected and interesting rules, but produces no spurious output.

6 Conclusions

In this work, we presented a novel method for mining association rules among node labels in a graph dataset. We relax the structural constraint used in subgraph mining, and discover rules that consist of sets of labels on both sides. A discovered rule $X \Rightarrow Y$ tells us that if we encounter all the labels in X in a tightly connected form somewhere in the input graph, there is a high probability that all the labels in Y will be encountered nearby. We evaluate the rules by looking at how far from itemset X, on average, do we need to look in order to find itemset Y. This approach provides more insight into the data than merely mining itemsets or subgraphs. We developed an algorithm for mining association rules, and experimentally confirmed its efficiency and usefulness. In future work, we intend to look at the possibility of extending this work to a setting where the dataset consists of multiple graphs, rather than a single graph.

7 Acknowledgments

This work was supported by the Fund for Scientific Research –Flanders (FWO-Vlaanderen) projects "Evolving graph patterns" and "Data mining for privacy in social networks".

References

[1] Abdulrehman, D., Monteiro, P.T., Teixeira, M.C., Mira, N.P., Lourenço, A.B., dos Santos, S.C., Cabrito, T.R., Francisco, A.P., Madeira, S.C., Aires, R.S., Oliveira, A.L., Sá-Correia, I., Freitas, A.T.: YEASTRACT: providing a programmatic access to curated transcriptional regulatory associations in Saccharomyces cerevisiae through a web services interface. Nucleic Acids Research 39 (2011)

[2] Agrawal, R., Imieliński, T., Swami, A.: Mining association rules between sets of items in large databases. In: Proc. of the ACM SIGMOD Int. Conf. on Management of Data. pp. 207–216 (1993)

[3] Cook, D.J., Holder, L.B.: Substructure discovery using minimum description length and background knowledge. Journal of Artificial Intelligence Research 1, 231–255 (1994)

[4] Cule, B., Goethals, B.: Mining association rules in long sequences. In: Proc. of the 14th Pacific-Asia Conf. on Knowledge Discovery and Data Mining. pp. 300–309 (2010)
[5] Dehaspe, L., Toivonen, H.: Discovery of frequent datalog patterns. Data Mining and Knowledge Discovery 3, 7–36 (1999)
[6] Hendrickx, T., Cule, B., Goethals, B.: Mining cohesive itemsets in graphs. In: Proc. of the 17th Int. Conf. on Discovery Science (2014)
[7] Hunter, S., Jones, P., Mitchell, A., Apweiler, R., Attwood, T.K., Bateman, A., Bernard, T., Binns, D., Bork, P., Burge, S., de Castro, E., Coggill, P., Corbett, M., Das, U., Daugherty, L., Duquenne, L., Finn, R.D., Fraser, M., Gough, J., Haft, D., Hulo, N., Kahn, D., Kelly, E., Letunic, I., Lonsdale, D., Lopez, R., Madera, M., Maslen, J., McAnulla, C., McDowall, J., McMenamin, C., Mi, H., Mutowo-Muellenet, P., Mulder, N., Natale, D., Orengo, C., Pesseat, S., Punta, M., Quinn, A.F., Rivoire, C., Sangrador-Vegas, A., Selengut, J.D., Sigrist, C.J.A., Scheremetjew, M., Tate, J., Thimmajanarthanan, M., Thomas, P.D., Wu, C.H., Yeats, C., Yong, S.Y.: InterPro in 2011: new developments in the family and domain prediction database. Nucleic Acids Research 40(D1), D306–D312 (Jan 2012)
[8] Huntley, R.P., Sawford, T., Mutowo-Meullenet, P., Shypitsyna, A., Bonilla, C., Martin, M.J., O'Donovan, C.: The goa database: gene ontology annotation updates for 2015. Nucleic Acids Research p. gku1113 (2014)
[9] Inokuchi, A., Washio, T., Motoda, H.: Complete mining of frequent patterns from graphs: Mining graph data. Machine Learning 50(3), 321–354 (2003)
[10] Inokuchi, A., Washio, T., Motoda, H., Kumasawa, K., Arai, N.: Basket analysis for graph structured data. In: Proc. of the 3rd Pacific-Asia Conf. on Methodologies for Knowledge Discovery and Data Mining. pp. 420–431 (1999)
[11] Kerrien, S., Aranda, B., Breuza, L., Bridge, A., Broackes-Carter, F., Chen, C., Duesbury, M., Dumousseau, M., Feuermann, M., Hinz, U., Jandrasits, C., Jimenez, R.C., Khadake, J., Mahadevan, U., Masson, P., Pedruzzi, I., Pfeiffenberger, E., Porras, P., Raghunath, A., Roechert, B., Orchard, S., Hermjakob, H.: The IntAct molecular interaction database in 2012. Nucleic Acids Research 40, D841–D846 (Jan 2012)
[12] Khan, A., Yan, X., Wu, K.L.: Towards proximity pattern mining in large graphs. In: Proc. of the 2010 ACM SIGMOD Int. Conf. on Management of Data. pp. 867–878 (2010)
[13] Licata, L., Briganti, L., Peluso, D., Perfetto, L., Iannuccelli, M., Galeota, E., Sacco, F., Palma, A., Nardozza, A.P., Santonico, E., Castagnoli, L., Cesareni, G.: MINT, the molecular interaction database: 2012 update. Nucleic Acids Research 40(D1), D857 –D861 (Jan 2012)
[14] Nijssen, S., Kok, J.: The gaston tool for frequent subgraph mining. Electronic Notes in Theoretical Computer Science 127, 77–87 (2005)
[15] Stark, C., Breitkreutz, B.J., Chatr-aryamontri, A., Boucher, L., Oughtred, R., Livstone, M.S., Nixon, J., Auken, K.V., Wang, X., Shi, X., Reguly, T., Rust, J.M., Winter, A., Dolinski, K., Tyers, M.: The BioGRID interaction database: 2011 update. Nucleic Acids Research p. gkq1116 (Nov 2010)
[16] Washio, T., Motoda, H.: State of the art of graph-based data mining. ACM SIGKDD Explorations Newsletter 5, 59–68 (2003)
[17] Yan, X., Han, J.: gspan: Graph-based substructure pattern mining. In: Proc. of the 2002 IEEE Int. Conf. on Data Mining. pp. 721–724 (2002)
[18] Yoshida, K., Motoda, H., Indurkhya, N.: Graph-based induction as a unified learning framework. Journal of Applied Intelligence 4, 297–316 (1994)