What is Frequent in a Single Graph?

Björn Bringmann and Siegfried Nijssen

Katholieke Universiteit Leuven
Celestijnenlaan 200 A, B-3001 Leuven, Belgium
{Bjoern.Bringmann,Siegfried.Nijssen}@cs.kuleuven.be

1 Introduction

Pattern mining has been studied in different types of data, starting from itemsets up to highly structured data such as relational data or hypergraphs. Usually the setting is such that a multiset of these structures is given and the aim is to find patterns that can be mapped onto at least a minimum number of the multiset members. This is normally called the *transactional* setting with prominent examples being basket analysis, or molecular fragment mining.

However, some graph databases are hard to represent in the transactional setting. For instance, the web, or any social network, is a single large graph that one may not wish to split up into small parts. The focus when analysing these networks is to find structural regularities or anomalies within such networks, rather then finding structural regularities common to a *set* of them. Complementary to the *graph-transactional* setting this is called the *single-graph* setting. The single-graph setting introduces interesting problems that to not appear in the transactional setting.

We will discuss an approach for mining patterns in single graphs and some of the problems that are associated.

2 The Support of a Pattern

A labeled graph $g = (\mathbb{V}_g, \mathbb{E}_g, \lambda_g)$ consists of a a set of nodes $\mathbb{V}_g$, a set of edges $\mathbb{E}_g \subseteq \mathbb{V}_g \times \mathbb{V}_g$ and a labeling function $\lambda_g : \mathbb{V}_g \cup \mathbb{E}_g \to \Sigma$ that maps each element of the graph to an element of the alphabet Σ.Let G_Σ be the set of all graphs over the alphabet Σ. We define support as $\sigma : G_\Sigma \times G_\Sigma \to \mathbb{N}$.

As stated before, the usual constraint employed in frequent pattern mining is minimum support. To enable an efficient search when using the minimum support constraint, the support measure needs to be anti-monotonic. This means that for any graph g that is a subgraph of p: $\sigma(g, D) \geq \sigma(p, D)$ has to hold, where D is the data graph. This requirement is quite easily upheld for the transactional setting, but turns out to be rather tricky for the *single-graph* setting. First, it is unclear what should be counted. Second, we have to make sure that the anti-monotonicity property is fullfilled for the support measure.

Occurrence of a Pattern Given a pattern $p = (\mathbb{V}_p, \mathbb{E}_p, \lambda_p)$ and a graph $g = (\mathbb{V}_g, \mathbb{E}_g, \lambda_g)$ we call each subgraph o of g that is isomorphic to p an occurrence of p. For each occurrence $o = (\mathbb{V}_o, \mathbb{E}_o, \lambda_o)$ there is a function $\varphi : \mathbb{V}_p \rightarrow \mathbb{V}_g$ mapping the nodes of the pattern to the according nodes in the graph such that (I) $\forall v \in \mathbb{V}_p \Rightarrow \lambda_p(v) = \lambda_g(\varphi(v))$ and (II) $\forall (u, v) \in \mathbb{E}_p \Rightarrow (\varphi(u), \varphi(v)) \in \mathbb{E}_g$.

We will explain the problem of the support measure on a single graph using the example shown in Figure 1. Pattern p_1 has one occurrence in graph g. Graph p_2 a specialisation of p_1. The question arises what the support of p_2 in g is. In a *transactional* setting it would be sufficient to evaluate if there is at least one occurrence of p_2 in g. In a *single graph* setting we need to count the support in the single graph. Intuitively there are eight possibilities to match p_2 onto g. Unfortunately this would violate the anti-monotonicity which we require for the support measure.

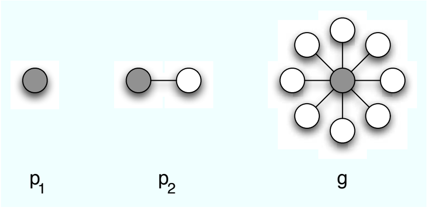

Fig. 1. The support problem in a single graph g: p_1 occurs once. How often occurs p_2?

Maximal Independent Set To solve this problem, Kuramochi *et al.* [2] introduced a support measure based on solving maximal independent set problems. For this measure, each possible occurrence o_i of a pattern p in the graph g is calculated. Then a *overlap*-graph is created. Each node in the overlap-graph corresponds to one occurrence o_i. In case that two occurrences o_j and o_k *share* an edge, there is an edge (o_j, o_k) in the overlap-graph. The support for the pattern p is then equal to the size of the maximal independent set of the overlap-graph. We denote this support measure by σ_s. Consider the example in Figure 2. There are three occurrences of the pattern p in the graph g. The occurrence in the 'middle' shares an edge with each of the other two occurrences which leads to the overlap-graph shown on the right. The maximal independent set of this overlap-graph has size two, thus the support of the pattern p is two according to the measure based on the maximal independent set.

It can be shown that this support measure is monotonic. However, solving a maximally independent set problem is NP-complete.

Most Restricted Node The *single node* based support measure we present here avoids potentially expensive maximal independent set computations. It is based on the number of unique nodes in $g = (\mathbb{V}_g, \mathbb{E}_g)$ that a node of the pattern $p = (\mathbb{V}_p, \mathbb{E}_p)$ is mapped to. Given a function $\varphi_i : \mathbb{V}_p \rightarrow \mathbb{V}_g$ for each occurrence o_i of p we define the support as:

$$\sigma_n(p, g) = \min_{v \in \mathbb{V}_p} |\{\varphi_i(v) : \varphi_i \text{ is a valid mapping}\}|$$

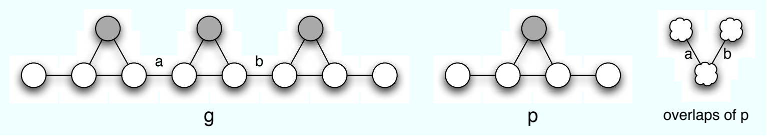

Fig. 2. Pattern p can be mapped at least three times to g. However, the occurrences overlap as shown on the right. The size of the maximal independent set of the overlap-graph is two.

By taking the node in p which is mapped to the least number of unique nodes in g, we can ensure the anti-monotonicity of σ_n. As the experiments show, the node-based support yields slightly more patterns than the support based on the maximal independent set. Figure 3 (left) shows an example of this behaviour. However, the other case could occur as well as depicted in Figure 3 (right). Please note that the support measure $\sigma_n$can easily be modified to count the minimum number of unique edges an edge of the pattern is mapped to. This would be an upper bound for σ_s.

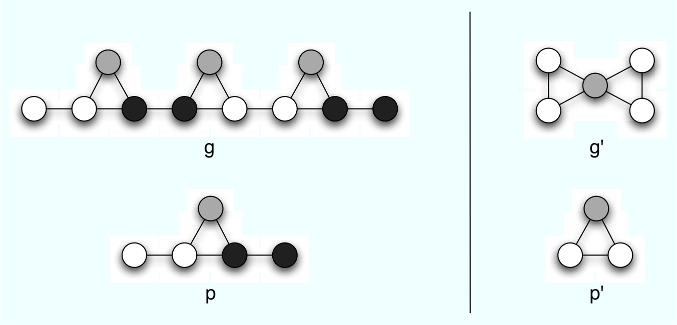

Fig. 3. An example similar to Figure 2 on the left with $\sigma_n(p, g) = 3$ and $\sigma_s(p, g) = 2$. On the right an example with $\sigma_n(p', g') = 1$ and $\sigma_s(p', g') = 2$.

3 Experiments

To evaluate the introduced support measure, we performed experiments on two different datasets. The mining-algorithm is based on a gSpan implementation [3] adapted for mining single graphs using the presented support measure σ_n. We used a Pentium IV with 3,2GHz and 2GB main memory. The results for the maximal independent set are taken from [2] and where executed on a AMD Athlon with 1.53 GHz and 2GB main memory.

One the *credit* dataset we observe that both measures yield the same number of patterns. It is very likely that the patterns found are the same. This might

	minimum	Patterns		Runtime (in seconds)	
Dataset	support	σ_s	σ_n	σ_s	σ_n
Credit	200	1325	1325	10	8
	100	11696	11696	45	46
	50	73992	73992	172	179
	20	613884	613884	1855	1129
Aviation	1750	2249	2255	787	124
	1500	5207	5231	1674	244
	1250	11087	11155	2720	534
	1000	30331	30457	5158	1352

Table 1. Comparing both support measures on different datasets for several different minimum support constraints.

be due to the rather *transactional* dataset wich consists of 700 connected components, 21 nodes and 20 edges each. For the *aviation* dataset we can see slight differences, but never more than 1%. Even given that the runtimes come from different systems, they indicate that the presented support measure is faster on this setting. Of course, this has to be investigated further.

4 Related Work

First the *SiGraM* algorithm introduced by Kuramochi *et al.* is developed for mining frequent patterns from a single graph. Since we work on the same setting, it was a natural choice to compare the results.

Furthermore several graph mining techniques were developed in the past years. The most popular among those is probably *gSpan* [3].All of those are designed for the *transactional* setting and find frequent patterns in a database given a minimum support threshold. The main difference among these approaches lies in the techniques employed to mine the patterns.

Finally *SubDue* [1] is an algorithm that can handle single graphs. However, its aim is not the extraction of *frequent* patterns, but to find substructures in the input that allow for efficient compression.

References

1. Lawrence B. Holder, Diane J. Cook, and Surnjani Djoko. Substucture discovery in the subdue system. In *KDD Workshop*, pages 169–180, 1994.
2. Michihiro Kuramochi and George Karypis. Finding frequent patterns in a large sparse graph. *Data Min. Knowl. Discov.*, 11(3):243–271, 2005.
3. Xifeng Yan and Jiawei Han. gspan: Graph-based substructure pattern mining. In *ICDM*, pages 721–724, 2002.