Frequent Graph Mining Based on Multiple Minimum Support Constraints[*]

Gangin Lee and Unil Yun[**]

Department of Computer Engineering, Sejong University, Republic of Korea
{abcnarak,yunei}@sejong.ac.kr

Abstract. Frequent graph mining allows us to find useful frequent sub-graph patterns from large and complicated graph databases. In a lot of real world applications, graph patterns with relatively low supports can be used as meaningful information. However, previous methods based on a single minimum support threshold have trouble finding them. That is, they cause "*rare item problem*", which means that useful sub-graphs with low supports cannot be extracted when a minimum support threshold is high, while an enormous number of patterns have to be mined to obtain these useful ones when the value is low. To overcome this problem, we propose a novel algorithm, FGM-MMS (Frequent Graph Mining based on Multiple Minimum Support constraints). After that, we demonstrate that the suggested algorithm outperforms a state-of-the-art graph mining algorithm through comprehensive performance experiments.

Keywords: Frequent graph mining, Multiple minimum supports, Sub graph.

1 Introduction

Graphs are useful data structure which can express almost all data from the real world effectively. Therefore, many studies for mining these graph data[3, 5, 6] have been conducted. Most graph databases obtained from the real world include not only high frequent sub-graph patterns but also meaningful ones with relatively low frequency(or support). However, since previous methods consider a single minimum support threshold, there occurs the following problem, i.e., *rare item problem*[4]. When the minimum support threshold is high, they cannot find meaningful sub-graph patterns with small supports while when it is low, they have to mine an enormous number of patterns to find the useful ones. In this paper, to solve the problem, we propose a novel graph mining algorithm applying multiple minimum support constraints, called FGM-MMS(Frequent Graph Mining based on Multiple Minimum Support constraints). The algorithm allows us to obtain useful sub-graphs without lowering the minimum support threshold more than needs. Experimental results provided in this

[*] This research was supported by the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (NRF No. 2012-0003740 and 2012-0000478).

[**] Corresponding author.

James J. (Jong Hyuk) Park et al. (eds.), *Mobile, Ubiquitous, and Intelligent Computing*,
Lecture Notes in Electrical Engineering 274,
DOI: 10.1007/978-3-642-40675-1_4, © Springer-Verlag Berlin Heidelberg 2014

paper show that the suggested algorithm has outstanding performance more than that of a state-of-the-art algorithm.

The remainder of this paper is organized as follows. In Section 2 and 3, we introduce related work and details of the proposed algorithm, FGM-MMS. In Section 4 and 5, we analyze results of our comprehensive experiments and conclude this paper.

2 Related Work

To solve the *rare item problem* from databases composed of itemsets, various frequent pattern mining approaches[1, 2, 4] have been proposed. MSApriori[4] is a level-wise method for solving this problem. A FP-growth-based algorithm, CFP-growth[1] solves the same issue more efficiently. CFP-growth++[2] is its advanced version. They conduct mining works considering different minimum item supports for each item, not a single value. Therefore, they can easily mine specific itemsets with low supports but important characteristics. However, since they are limited to mining itemset-based simple databases, they do not deal with complicated graph databases.

Among many fundamental graph mining algorithms, Gaston[5, 6], which is a state-of-the-art algorithm, performs the most efficient mining operations. The algorithm extracts frequent sub-graph patterns classifying its mining procedure into the three phases: *path*, *free tree*, and *cyclic graph*. In addition, it uses *embedding list* to reduce runtime needed for mining patterns. Our algorithm is compared with the Gaston for fair performance comparison. To our knowledge, FGM-MMS is the first graph mining algorithm which can solve the *rare item problem*. By applying the multiple minimum support constraints, it can extract meaningful sub-graph patterns considering supports for each element of graphs.

3 Mining Frequent Sub-graph Patterns Based on Multiple Minimum Support Constraints

3.1 Applying Multiple Minimum Support Constraints on the Graph Mining

In contrast to itemsets composed of only items, graph patterns consist of a number of elements such as vertices and edges. Therefore, the graph mining approach should consider all the elements. Furthermore, in this multiple minimum support framework, we have to consider different minimum element support values for each element.

Definition 1. (*MES*(*Minimum Element Support*)) Let $V = \{v_1, v_2, \ldots, v_m\}$ be a set of vertices in any graph, G, and $E = \{e_1, e_2, \ldots, e_n\}$ be a set of edges in G. Then, *MES* values for each element in V and E can have user-specified values or other ones discussed in [4]. $MES(v_k)$ and $MES(e_l)$ mean *MES* values for v_k and e_l respectively. If a support of any element x, $SUP(x)$ is lower than $MES(x)$, x becomes useless one.

Graph patterns are composed of a number of the elements, and each element has its own *MES* value. General graph mining methods determine with a given minimum support threshold whether any graph pattern is frequent or not. However, in this paper, we use a different criterion to solve the *rare item problem*.

Definition 2. (*MGS*(*Minimum sub-Graph Support*)) Given a sub-graph, G, its $V = \{v_1, v_2, \ldots, v_i\}$, and $E = \{e_1, e_2, \ldots, e_j\}$, then, $MGS(G) = minimum(MES(v_1), \ldots, MES(v_i), MES(e_1), \ldots, MES(e_j))$. *MGS* is used to check whether G is valid or not. Thus, if $SUP(G)$ is smaller than $MGS(G)$, G becomes an invalid sub-graph pattern.

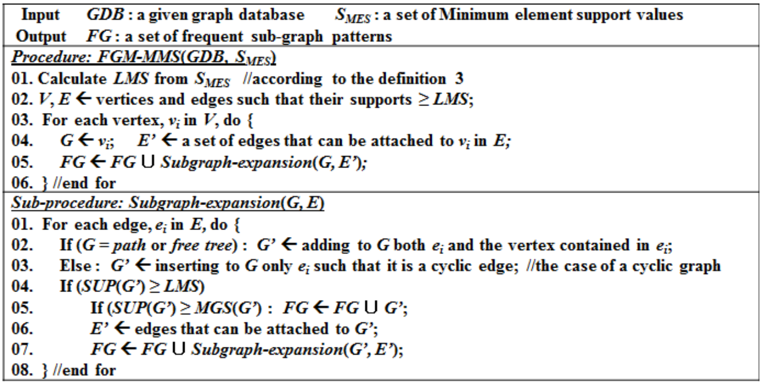
The reason why *MGS* is computed in this way is to consider the rarest element among the graph. Therefore, a certain sub-graph satisfying the *MGS* condition becomes a valid pattern with useful rarity. However, if we directly delete elements or sub-graphs which do not satisfy the *MES* or *MGS* conditions, fatal pattern losses can be caused since pruning by them does not satisfy the anti-monotone property. That is, given a sub-graph with invalid elements, although it is currently infrequent, certain supper patterns of it may be frequent as its graph expansion works are performed. In all of the pattern mining areas, maintaining this property is one of the most important factors since we can prune invalid elements or sub-graph patterns in advance only when that property is effective. For this reason, we need a way to maximize pruning efficiency maintaining that property.

Definition 3. (*LMS*(*Least Minimum Support for the graph mining*)) *LMS* is a pruning factor without causing any pattern loss. Given a set of *MES*, S_{MES}, it is sorted in *MES* descending order, denoted as $S_{MES} = \{MES(x_i), MES(x_j), \ldots, MES(x_l)\}(\forall x \mid x \in V$ or E, $MES(x_i) \geq MES(x_j)\ldots\geq MES(x_l))$. After that, Starting from x_l, we compare $SUP(x_l)$ with $MES(x_l)$. If there exists any element, x_k, satisfying $SUP(x_k) \geq MES(x_k)$, $MES(x_k)$ becomes *LMS*. Pruning works by *LMS* maintains the anti-monotone property.

Certain elements having lower supports than *LMS* do not make valid sub-graphs in any case since all possible super patterns expanded from them have also lower values than it. Likewise, if a certain graph has a smaller support than *LMS*, its all of the possible supper sub-graph patterns also have lower values than *LMS*. Accordingly, they can be pruned in advance without causing any problem. That is, the anti-monotone property is satisfied. In addition, by pruning those useless ones, we can reduce runtime and memory resources needed for the mining operations effectively.

3.2 FGM-MMS Algorithm

Fig. 1 presents an overall mining procedure of our FGM-MMS. In the procedure: *FGM-MMS*, the algorithm first calculates *LMS* according to the definition 3 and then finds valid V and E(lines 1~2). After that, it generates valid sub-graphs considering the multiple minimum support constraints, performing expansion operations for graphs(lines 3~6). When the sub-procedure, *Subgraph-expansion* is called, the algorithm conducts the graph expansion works with different ways depending on current graph forms(lines 2~3). If a support of the expanded graph, G', $SUP(G')$ is lower than *LMS*, it is permanently removed. Otherwise, the support is compared with $MGS(G')$ again, where G' is inserted in FG if its support is not smaller than $MGS(G')$(line 5). Thereafter, the algorithm performs mining operations expanding G' recursively(lines 6~7). After all of the expansions terminate, we can gain the complete FG considering the multiple minimum support constraints.



```
Input    $GDB$ : a given graph database    $S_{MES}$ : a set of Minimum element support values
Output   $FG$ : a set of frequent sub-graph patterns

Procedure: FGM-MMS(GDB, $S_{MES}$)
Calculate $LMS$ from $S_{MES}$   //according to the definition 3
$V$, $E$ ← vertices and edges such that their supports $\geq LMS$;
For each vertex, $v_i$ in $V$, do {
    $G$ ← $v_i$;   $E'$ ← a set of edges that can be attached to $v_i$ in $E$;
    $FG$ ← $FG \cup$ Subgraph-expansion($G$, $E'$);
} //end for

Sub-procedure: Subgraph-expansion(G, E)
For each edge, $e_i$ in $E$, do {
    If ($G$ = path or free tree) :  $G'$ ← adding to $G$ both $e_i$ and the vertex contained in $e_i$;
    Else :  $G'$ ← inserting to $G$ only $e_i$ such that it is a cyclic edge; //the case of a cyclic graph
    If ($SUP(G') \geq LMS$)
        If ($SUP(G') \geq MGS(G')$ :  $FG$ ← $FG \cup G'$;
        $E'$ ← edges that can be attached to $G'$;
        $FG$ ← $FG \cup$ Subgraph-expansion($G'$, $E'$);
} //end for
```

Fig. 1. FGM-MMS algorithm

4 Performance Analysis

We compare our FGM-MMS with Gaston[5, 6] for objective performance evaluation since the Gaston, as a state-of-the-art algorithm, has more outstanding performance than the others. They were written in C++ and ran with 3.33GHz CPU, 3GB RAM, and WINDOWS 7 OS. We use a real graph dataset, named DTP[6], to show how effectively our FGM-MMS is performed on the real-world graph data.

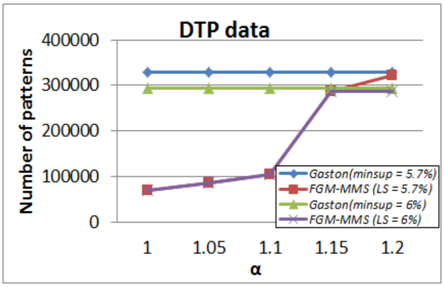


Fig. 2. The number of frequent sub-graph patterns

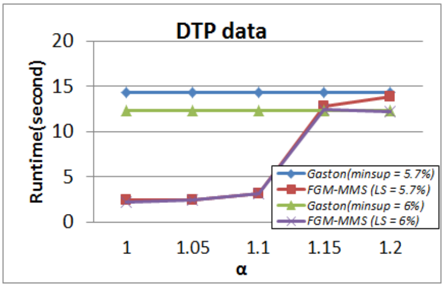


Fig. 3. Runtime result

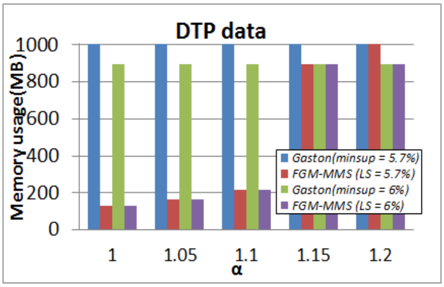


Fig. 4. Memory usage

To assign *MES* for each element, we applied the methodology discussed in [4]. Its formula is $MES(e_k) = maximum(\beta * SUP(e_k), LS)$, where $SUP(e_k)$ means a support of any element, e_k, and *LS* is the user-specified lowest *MES* and has the same value as the minimum support threshold used in Gaston. $\beta(0\leq \beta \leq 1)$ represents how closely *MES* for a certain element is related to its support. As the value becomes closer to 1, relevance becomes higher. It is denoted as $\beta = 1/\alpha$, and α is variable.

In Fig. 2 to 4, we can observe that FGM-MMS shows outstanding performance more than that of Gaston in all cases when α is relatively low. The reason is as follows. When α is high, *MES* values are more likely to be set as *LS*, so in this case, our algorithm operates similarly to Gaston. Meanwhile, at lower values of α, FGM-MMS can better reflect support values for each element to their own *MES*, which means that the effect of the multiple minimum supports becomes larger. Thus, the proposed algorithm can extract valid sub-graph patterns with smaller runtime and memory resources as shown in the figures.

5 Conclusion

In this paper, we proposed a frequent graph mining algorithm based on multiple minimum support constraints, called FGM-MMS. In contrast to the previous algorithms using a single minimum support threshold, our algorithm could solve the challenging issue, *rare item problem* on the graph mining area by using the multiple minimum support constraints, thereby mining more meaningful frequent sub-graph patterns. Furthermore, we demonstrated that the proposed algorithm shows more outstanding performance compared to the previous state-of-the-art algorithm through comprehensive performance experiments provided in this paper.

References

1. Hu, Y.H., Chen, Y.L.: Mining association rules with multiple minimum supports: a new mining algorithm and a support tuning mechanism. Decision Support Systems 42(1), 1–24 (2006)
2. Kiran, R.U., Reddy, P.K.: Novel techniques to reduce search space in multiple minimum supports-based frequent pattern mining algorithms. In: 14th International Conference on Extending Database Technology, pp. 11–20 (2011)
3. Lee, G., Yun, U.: Mining weighted frequent sub-graphs with weight and support affinities. In: Proceedings of the 6th Multi-Disciplinary International Workshop on Artificial Intelligence, pp. 227–238 (2012)
4. Liu, B., Hsu, W., Ma, Y.: Mining association rules with multiple minimum supports. In: Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 337–341 (1999)
5. Nijssen, S., Kok, J.N.: The Gaston Tool for Frequent Subgraph Mining. Electronic Notes in Theoretical Computer Science 127(1), 77–87 (2005)
6. Nijssen, S., Kok, J.N.: A quickstart in frequent structure mining can make a difference. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 647–652 (2004)