Journal of Theoretical and Applied Information Technology
30th April 2015. Vol.74 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

IDFP-TREE: AN EFFICIENT TREE FOR INTERACTIVE MINING OF FREQUENT SUBGRAPH PATTERNS

MOHAMMAD H. NADIMI-SHAHRAKI, MARYAM TAKI, MOHAMMAD NADERI

Faculty of Computer Engineering, Najafabad branch, Islamic Azad University, Najafabad, Iran
E-mails: nadimi@iaun.ac.ir , maryam.taki@sco.iaun.ac.ir , naderi@iaun.ac.ir

ABSTRACT

Recently, knowledge extraction from transactional graph databases by mining frequent subgraph patterns has become an interesting research topic. One of the important challenges in this topic is the situation called interactive mining in which the minimum support threshold can be frequently changed to find proper frequent subgraph patterns. Obviously, running the mining method from scratch is very costly. Therefore, in this paper, an efficient tree called IDFP-tree is proposed to develop our previous two-layer model for interactive mining of frequent subgraph patterns from transactional graph databases. The proposed tree constructs the mining model separated from the mining process. Thus, when the minimum support threshold is changed, there is no need to reconstruct the mining model, and only the mining process must be rerun. The experimental results show when the mining model is constructed by IDFP-tree, it can be frequently used and the total runtime of interactive mining of frequent subgraph patterns can be reduced.

Keywords: *Interactive Mining, Graph Mining, Transactional Graph, Frequent Subgraph Pattern.*

1. INTRODUCTION

Nowadays, an increasing number of large transactional graph databases have been created. This is because; graph is a suitable structure to represent structured and semi-structured data of applications such as bioinformatics, chemistry and social networks [1]. Recently, many studies have been conducted to discover valuable but hidden knowledge in these databases through using data mining techniques. Consequently, numerous methods have been introduced for mining frequent subgraph patterns [2-9]. The main challenge facing all studies carried out in graph mining is the presence of massive databases that need to be constantly scanned [2]. This challenge is very conspicuous in discovering frequent subgraph patterns especially in interactive environments because it makes the mining process very costly. If frequent subgraph patterns can be updated interactively without rescanning databases, it will prevent waste of time. Most of algorithms for frequent subgraph pattern mining are based on two main methodologies Apriori and Pattern-growth. The most of proposed algorithms have been developed for static mining. However, in some applications, users need to change new minimum support threshold interactively to find new and proper frequent patterns. Application of graphs in dynamic environments and low efficiency in previous methods of interactive frequent subgraph mining is caused that introduction an interactive frequent subgraph mining method would be necessary. Nadimi-Shahraki et al. [10] discussed this problem and proposed a two layer model for interactive mining. In this paper, based on their proposed model, an efficient tree called IDFP-tree is proposed for interactive mining of frequent subgraph patterns. IDFP-tree is built in the form of a compact and complete tree by once scanning the graph database in which the mining model can adapt itself to graph structure. The experimental results show that by using IDFP-tree, there is no need to reconstruct the mining model when user changes the minimum support threshold. The rest of the paper is organized as follows. In the next section, the related works are reviewed. The two-layer model is discussed in Section 3. Then, the proposed tree is introduced in Section 4. Then, the evaluation results and conclusions are presented in Sections 5 and 6 respectively.

2. RELATED WORKS

There have been proposed many algorithms for frequent subgraph pattern mining based on two general methodologies Apriori and Pattern-growth [11]. Similar to frequent pattern mining, Apriori based algorithms for frequent subgraph pattern mining must combine the frequent k-subgraph patterns extracted from the previous level to

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

generate K+1-subgraph candidates. Therefore, k database scans are needed for mining of frequent k-subgraph patterns [11] which is very costly especially when the graph database is very large.

Consequently, there have been introduced following three Apriori-based algorithms: AGM [2], FSG [3] and PM [4]. They generate candidate patterns based on edges, nodes and unconnected paths respectively which is the main difference between these algorithms. Inherently, all of these algorithms suffer from same weaknesses. They need a large amount of memory because of employing breath search methods and generate a large volume of candidate patterns, although they make use from pruning techniques. Consequently, many database scans and examine a large number of candidates are very costly. Furthermore, these algorithms are not adoptable to interactive mining [10].

To solve these weaknesses, J. Han et al. proposed FP-growth in 2000 [14]. It aims to enhance the efficiency of frequent pattern mining by using a novel tree structure called FP-tree. Efficiently, data is captured and kept by FP-tree by two database scans. Then, to mine frequent patterns, FP-growth algorithm explores the tree in a divide-and-conquer strategy without candidate pattern generation. Consistently, a few efficient algorithms have been proposed based on pattern-growth for frequent subgraph pattern mining. However, these algorithms are not easily adaptable with interactive mining. This is because, they are developed based on an integrated and single-layer model which cannot be used for interactive mining. One of the most popular of these algorithms is gSpan that was introduced by Han and Yan in 2002 [6]. Inherently, this algorithm cannot be adapted with interactive mining and it must be rerun from scratch

Fortunately, a few interactive algorithms have been proposed for frequent pattern mining from which we try to get idea for developing an interactive mining algorithm for frequent subgraph pattern mining.

In 2002, R.P. Gopalan introduced DB-tree algorithm [15] where both frequent and infrequent are kept in the tree. Thus, this algorithm is easily adoptable to interactive mining of frequent patterns. In 2007, C.K.S Leung proposed a new tree structure called Can-Tree [16] for incremental and interactive mining. It captures contents of a transactional database by one database scan. The items in a Can-tree are sorted based on a standard and convectional order and inserted in a tree. In 2009, a new tree structure called CP-tree was introduced by S.K Tanbeer et al. [17]. It keeps the content of a database by scanning once and then uses an FP-growth algorithm for mining frequent patterns. CP-tree is easily adoptable to interactive mining. Then, in 2012 DFP-tree [18] was introduced which revises the process of building the CP-tree and thereby offers a tree for interactive and enhanced mining that is built by one scan of the database.

In next section, using the ideas of the above algorithms and the two-layer model [10] introduced for interactive mining of frequent pattern mining, an efficient tree for interactive mining of frequent subgraph pattern mining is proposed.

3. THE TWO-LAYER MODEL FOR INTERACTIVE MINING

The most efficient algorithms for frequent subgraph pattern mining are based on an integrated single-layer model in which the mining model and mining process are mixed. Mostly, in these methods, the process is done regarding to an adjusted minimum support threshold by which only frequent subgraphs are kept. Thus, this integrity makes these algorithms unable to be used for interactive mining of frequent subgraph patterns. The interactive mining is based on "build once, mine many" [8] and to support this principle for interactive mining, the mining model construction must be separated from the mining process [10]. Consequently, a two-layer model for interactive mining was proposed [10] and the experimental results show that by using this model there is no need to reconstruct the mining model when user changes the minimum support threshold. As shown in figure 1, in the first layer the mining model is constructed and the mining materials are formed. In the mining model layer, attempts are made to offer a structure that is as optimal as possible to be used in the mining process layer. Therefore, the database is captured and kept in memory by an optimal structure and then it can be frequently mined by changing the minimum support threshold [10].

4. THE PROPOSED TREE

In this section, based on the two-layer model explained in previous section, an efficient tree called IDFP-tree is proposed for interactive mining of frequent subgraph patterns. IDFP-tree is an extension of DFP-tree [18] which introduced for

Journal of Theoretical and Applied Information Technology
30th April 2015. Vol.74 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

frequent pattern mining. It aims to construct mining model separated from the mining process.

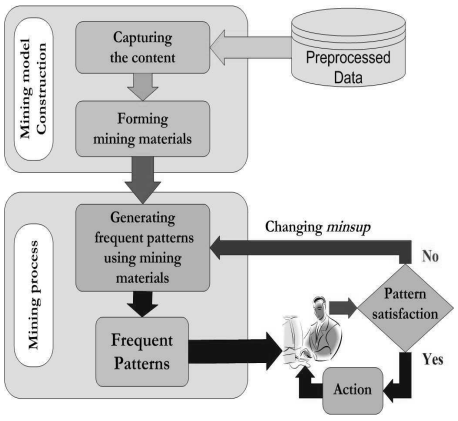

Figure 1: Two-Layer Model For Interactive Mining [10].

Thus, there is no need to reconstruct the mining model when user changes the minimum support threshold. As Figure 2 showed the algorithm of IDFP-tree, it is a compact structure composed of a graph database that is created by scanning the database once, and is built in the two main stages of embedding and reconstruction.

Algorithm: **IDFP-tree**. *Create Index base Dynamic FP tree*
Input: *Graph Index list*
Output: *IDFP-tree*

```
1. create the root of IDFP-tree and label it as "Null"
2.create tempheadertable
3. foreach Index ti in Graph Index list
4. {
5.    foreach item I in ti.item
6.        tempheader.Add(I)
7.    tempheadertable.sort()
8.    transaction.sort(tempheadertable)
9.    transaction ti insert into IDFP-tree
10.}
```

Figure 2: A pseudo code of the proposed tree.

Consistently, in the embedding stage, the IDFP-tree is built or updated based on the last frequency of each existing index. This stage is completed by reading the database only once. Reconstruction of the tree changes the semi-ordered existing tree into the format of a compact indexed FP-tree. In the mining process also, the efficient mining algorithm FP-growth is used for exploring the tree [18]. Figure 3 shows an IDFP-tree built on the basis of indices in Table 1.

Table 1: Table Of Converted Graphs From G0 To G3

Graphs	Index
MG1	I1,I2,I3,I4
MG2	I2,I3,I4
MG3	I0,I3
MG4	I0,I1,I3

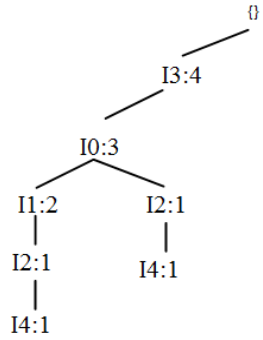

Figure 3: An IDFP-Tree For G0 To G3 Graphs

5. EVALUATION

In this paper, the efficiency of using IDFP-tree is compared with gSpan as the most popular algorithm for frequent subgraph pattern mining. The proposed tree and gSpan algorithm are experimentally compared conducted by four different datasets. Firstly, they are evaluated for mining model construction or static mining where the minimum support threshold is fixed. The experimental results shown in Figure 4 indicate the mining model construction by using the proposed model is faster than using gSpan algorithm. Since gSpan is based an integrated model, it is executed from scratch when the minimum support threshold is changed. Thus, it is expected the performance of proposed tree is more than IDFP-tree for interactive mining. Consistently, in each experiment, the algorithms are separately run in the same experimental environment.

In second experiment, the efficiency of using IDFP-tree for interactive mining is evaluated. In the mining model constructed by using IDFP-tree, only the mining stages were carried out from the beginning when the minimum support threshold is changed. However, in the gSpan algorithm all the stages must be rerun from the scratch. The experimental results shown by Figure 5, present the proposed tree is more efficient than gSpan for interactive mining of frequent subgraph patterns. IDFP-tree constructs the mining model independent

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

of the minimum support threshold, thus it fits "build once, mine many" principle and there is no need to reconstruct the mining model when the minimum support threshold is changed.

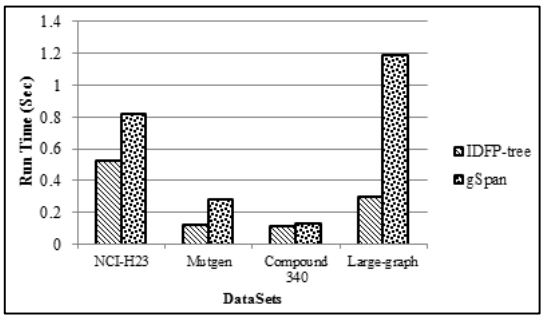

Figure 4 : Run Time Of Mining Model Construction By Using The Proposed Tree And The Gspan Algorithm.

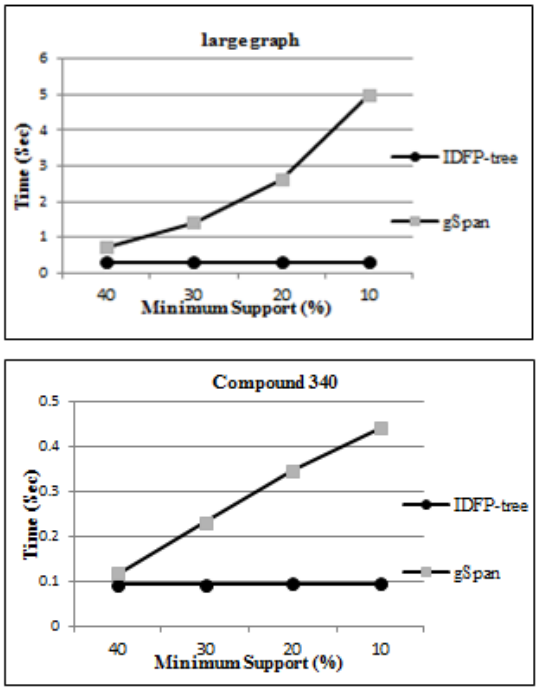

Figure 5: Runtime Of Interactive Mining By Using The Proposed Tree And Gspan

6. CONCLUSION

Recently, graph mining especially frequent pattern mining has become an interesting research topic. On the other hand, in many applications, interactive mining is a need where user changes minimum support threshold to find proper frequent patterns. In this paper, an efficient tree called IDFP-tree was proposed to develop the two-layer model [10] for interactive mining of frequent subgraph patterns. Using the proposed tree separates the mining model from the mining process. Therefore, only the mining process is rerun when the minimum support threshold is changed and there is no need for mining model reconstruction. Optimal structure to mining model layer is very important to increase performance of this method. Therefore it is a popular research to this field. The experimental results show that mining model construction by using the proposed tree is faster than using gSpan. Moreover, the run time of interactive mining of frequent subgraph patterns by using the proposed tree is smaller than using gSpan.

REFRENCES:

[1] Wang, Haixun. Managing and mining graph data. Ed. Charu C. Aggarwal. Vol. 40. New York: Springer, 2010

[2] Jiang, Chuntao, Frans Coenen, and Michele Zito. "*A survey of frequent subgraph mining algorithms*." The Knowledge Engineering Review 28.01 (2013): 75-105.

[3] Kuramochi, Michihiro, and George Karypis. *"An efficient algorithm for discovering frequent subgraphs."* Knowledge and Data Engineering, IEEE Transactions on 16.9 (2004): 1038-1051.

[4] Gudes, Ehud, Solomon Eyal Shimony, and Natalia Vanetik. *"Discovering frequent graph patterns using disjoint paths*." Knowledge and Data Engineering, IEEE Transactions on 18.11 (2006): 1441-1456.

[5] Borgelt, Christian, and Michael R. Berthold. *"Mining molecular fragments: Finding relevant substructures of molecules*." Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on. IEEE, 2002.

[6] Yan, Xifeng, and Jiawei Han. "gspan: Graph-based substructure pattern mining." Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on. IEEE, 2002.

[7] Wang, Chen, et al. *"Scalable mining of large disk-based graph databases*." Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2004.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[8] Huan, Jun, Wei Wang, and Jan Prins. *"Efficient mining of frequent subgraphs in the presence of isomorphism."* Data Mining, 2003. ICDM 2003. Third IEEE International Conference on. IEEE, 2003.

[9] Nijssen, Siegfried, and Joost N. Kok. "*A quickstart in frequent structure mining can make a difference*." Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2004.

[10] Nadimi-Shahraki, M. H., and Norwati Mustapha. *"A Two-layer Model for Interactive Mining of Frequent Patterns*." Proceedings of the International MultiConference of Engineers and Computer Scientists. Vol. 1. 2011.

[11] Lakshmi, K. "Frequent subgraph mining algorithms-a survey and framework for classification." (2012).

[12] Agrawal, Rakesh, and Ramakrishnan Srikant. *"Fast algorithms for mining association rules."* Proc. 20th int. conf. very large data bases, VLDB. Vol. 1215. 1994.

[13] Zaki, Mohammed Javeed. *"Scalable algorithms for association mining."* Knowledge and Data Engineering, IEEE Transactions on 12.3 (2000): 372-390.

[14] Han, Jiawei, Jian Pei, and Yiwen Yin. *"Mining frequent patterns without candidate generation."* ACM SIGMOD Record. Vol. 29. No. 2. ACM, 2000.

[15] Cheung, William, and Osmar R. Zaiane. *"Incremental mining of frequent patterns without candidate generation or support constraint."* Database Engineering and Applications Symposium, 2003. Proceedings. Seventh International. IEEE, 2003.

[16] Leung, Carson Kai-Sang, et al. *"CanTree: a canonical-order tree for incremental frequent-pattern mining."* Knowledge and Information Systems 11.3 (2007): 287-311.

[17] Tanbeer, Syed Khairuzzaman, et al. *"Efficient single-pass frequent pattern mining using a prefix-tree."* Information Sciences 179.5 (2009): 559-583.

[18] M. Hamedanian, M. Nadimi, and M. Naderi, "*An Efficient Prefix Tree for Incremental Frequent Pattern Mining*," International Journal of Information, vol. 3, 2013