A New Algorithm for Mining Frequent Itemsets Based on Fp-Search Algorithm with K Road Pruning

Hao JIANG
Computer Science and Engineering College
Southeast University
Nanjing, China
hjiang@seu.edu.cn

Ruda SHEN
Computer Science and Engineering College
Southeast University
Nanjing, China
shenruda3927@163.com

***Abstract*—Association rule mining is an important approach in data mining. Based on analyzing many previous algorithms such as Apriori, Fp-growth, Eclat and Fp-search, we propose a new algorithm named FPNMP-search to mine frequent itemsets. With no need to construct the MP-tree, FPNMP-Search algorithm can effectively prune the redundant path and mine all frequent itemsets. The experimental results show that FPNMP-search is more efficient than Fp-growth and Fp-search.**

Keywords—association rule mining;Fp-search; FPNMP-search; MP-tree

I. INTRODUCTION

At present, the main association rule mining algorithms is divided into three major categories: Apriori-like algorithm [1-2], Fp-growth-like algorithm [3], and Eclat-like algorithm [4]. These algorithms have the following problems: Apriori algorithm needs to scan the database many times that leading a large of I/O operations, and generate many redundant candidate sets. For Fp-growth algorithm, frequently constructing the conditional pattern base costs a lot of time. When the support threshold is relatively small, the compression capacity of Fp-tree begins to decrease, which leads to many unnecessary construction operations. For the large transaction database, the data represented by vertical data format of Eclat algorithm cannot be stored in memory and when the number of transaction is large, the size of tid-set will be large, the cost of the intersection of tid-set will increase, thus the performance of the algorithm will be reduced;

By using PE_Graph for the fast path extension, Tid-tree for compressing the tid-set and MP-tree for pruning the path, Fp-search algorithm can overcome the above problems and efficiently mine all frequent itemsets. But it still has shortcomings that when constructing the MP-tree and traversing the MP-tree, some extra time is needed [5]. In this paper, based on Fp-search algorithm, a new algorithm named FPNMP-search algorithm is proposed for efficiently mining all frequent itemsets.

This work was supported by Collaborative Innovation Center of Wireless Communications Technology and the National Natural Science Foundation of China (Grant No. 6157060279).

II. PROBLEM DESCRIPTION

Association rule mining means mining potentially valuable links between data items from a large number of data items. In general, association rule mining can be divided into two sub-problems: frequent itemsets generation and rule generation.

A. Frequent itemsets

Frequent itemsets mining is an important step in association rule mining. When generating all frequent itemsets, association rule mining is relatively easier. In fact, in recent years, the mining of association rules is mainly focused on mining frequent itemsets. Also, the main work of this paper is concentrated in mining frequent itemsets.

Let $i_k$representstransaction itemsets, which numbered TID. D represents the set of all transaction itemsets in the database. Let $i_k$represents data items. *I* denotes all data items in database D.

For any set X, if X is a subset of *I*, then X will be called Item set. The relationship between them is shownas follows.

$$I = \{i_1, i_2, \dots, i_m\} t_k = (TID, X) \mathrm{D} = \{t_1, t_2, \dots, t_n\}$$

For any itemset Y, the number of data items included in the item set is known as the dimension of item set Y, denoted | Y |.

Definition 1 For anyt_k, if Y is contained in X, then Y comprisest_k. The number of transaction itemsets that contain Y in the database D is called the absolute support of item set Y, denoted $\sigma(Y)$ .The ratio between the number of times Y appears in D and the total number of transaction itemsets in D is named the relative support of item set Y, denoted $\mathrm{support}(Y)$. The mathematical expressions of absolute support and relative support are showed as follows [6].

$$\sigma(Y) = |\{t_i | X \subseteq t_i \wedge t_i \in D\}|$$

$$\mathrm{support}(Y) = \sigma(Y) \,/\, |D|$$

Frequent itemsets mining, also known as frequent patterns mining, is a mining techniques for mining the interesting patterns from a large number of data. Whether a pattern is interesting or not, generally there are two criteria as follows [7]:

(1) Potential: When the pattern can make the user feel surprised, and could potentially provide a new message.

(2) Feasibility: Users can use this pattern to achieve the initial goal, and the pattern has actual practical value.

Definition 2 Supposing min_sup represents the minimum support threshold, if $\text{support}(Y) \geq \text{min\_sup}$, then Y is a frequent itemset. The collection of all frequent itemsets is denoted L. Frequent itemsets of dimension k are called k frequent itemsets. The range of the minimum support threshold min_sup is$[0,1]$, and the range of the minimum support count min_count is $[0, |D|]$, where $\text{min\_count} = \text{min\_sup} \times |D|$.

Let D be a transaction set database, I be the collection of all items and the minimum support threshold be min_sup, then the frequent itemsets L can be expressed as [8]:

$$L(D, \text{min\_sup}) = \{X \subseteq I | support(X, D) \geq \text{min\_sup}\}$$

B. *Association rule*

Association rule first appeared in the market basket analysis. A very classic example is the story "beer and diaper" from Wal-Mart. Wal-Mart's salesmen have found the fact that young father would buy a few bottles of beer while he went to buy diapers. This rule can be represented by a formula:

$$\text{diaper} \Rightarrow \text{beer}$$

The rule indicates the intrinsic link between the diapers and beer sales. In simple terms, this link is the association rule.

Association rule is a rule shaped like$X \Rightarrow Y$, where X and Y is a subset of I, and$X \cap Y = \Phi$. The rule means that if a transaction item contains X, it will contain Y. The total number of transaction items that contain $X \cup Y$is called the support count of the rule. The ratio between the support count and the total number of transaction itemsets in D is called support of the rule, also known as relative support, denoted $\text{support}(X \Rightarrow Y)$. The value of the support denoted $\text{P}(X \cup Y)$indicates the probability that X and Y appear in a transaction item of database D in the same time.

Definition 3 Confidence is an important method to measure whether the rule is effective. It represents the probability that Y may appear when X is in the transaction items, denoted $\text{confidence}(X \Rightarrow Y)$, and valued $\text{P}(Y|X)$. In order to show the support and confidence of the rules more clearly, see the following mathematical representation [9]:

$$support(X \Rightarrow Y) = P(X \cup Y)$$

$$confidence(X \Rightarrow Y) = P(Y|X)$$

Actually, association rule problem is the problem that all association rules are generated by two fitting parameters the minimum support threshold min_sup (range [0,1])and minimum confidence threshold min_conf (range [0, 1]). These two parameters are used to measure the strength of the relationship between the cause and effect in a rule (such as, in the rule$X \Rightarrow Y$, X is the cause and Y is the effect.). For example, when the frequency that XY exist in D in the same time is quite high, we have reasons to believe that there is a relationship between X and Y. Furthermore, if the probability that Y may appear when X is in the transaction items is quite high, which strengthen the credibility of the relationship between X and Y, otherwise the risk of misjudging the relationship is high.In general, the rules that meet the minimum support threshold and the minimum confidence threshold are strong rules.

C. *Association rules mining*

Association rule mining problem is the problem that find the intrinsic link between the items that meet the minimum support threshold and the minimum confidence threshold which are set by user from the large-scare database. Generally speaking, it is divided into two steps [6]:

Step 1 (generating all frequent itemsets): In this step, all itemsets that meet the minimum support threshold are found. This step is an important part of association rules mining, while the selection of the minimum support threshold is quite important. The main task in this step is to enhance the speed of finding the frequent itemsets.

Step 2 (generating the strong rules): Based on the frequent itemsets found in the Step 1, all the strong rules we generate are the association rules we want to find. For instance, for any item set X and Y, if Y is a subset of X and$support(X) / support(X - Y) \geq \text{min_conf}$, $X - Y \Rightarrow Y$ is a valid rule.

Association rule mining mainly includes two steps above. But the first step is far more difficulty than the second step, which is the major challenge of association rule mining. This paper mainly discusses the improvement of the first step.

Definition 4 Given a set of item set I, a transaction itemsets database D and I is the collection of all the sets in D. While setting appropriate minimum support threshold min_sup, the issue of access to$L(D, \text{min_sup})$ is the issue of frequent itemsets mining.

Frequent itemsets mining is an extremely issue. For the item set I, if$|I| = n$, the number of subsets of I is 2^n, where a lot of subset's support will be less than the minimum support threshold. If using exhaustive method to search the frequent itemsets, the cost and the time complexity will be too high. So it is necessary to find more efficient algorithm for mining frequent itemsets.

III. FPNMP-SEARCH ALGORITHM

Based on analyzing the Fp-search algorithm with K road pruning, we find that the performance and running time of Fp-search algorithm are quite well, when$k = 1,2,3$. However, there is a noteworthy and urgentproblem that constructing MP-tree in Fp-search algorithmwith K road pruning costs a lot of time and space, when the number of frequent itemsets is huge. Further analysis of the experimental results, we find that when the PE-Graph is dense (frequent itemsets are high density), the performance of Fp-search algorithm with K road pruning is equal to the Fp-search algorithm with 0 road pruning. The reason of this phenomenon is that when the PE-Graph is dense, the number of paths can be pruned is finite (the number of redundant paths is less). The calculation time reduced by pruning is equal to or even less than the consumption time of pruning itself. At the same time, due to the large number of frequent itemsets, the space complexity of constructing MP-tree is high and pruning redundant paths costs a lot of memory.

When$k = 1,2,3$, the performance of Fp-search algorithm with K road pruning is good and stable. By the mathematical proofs, the frequency of Fp-search algorithm with K road pruning is a minimum of $1 - 1 / 2^{k+1}$. When $k = 1$, the frequency is 75%and the performance of the algorithm is as good as$k = 2,3$. In other words, if we can find an algorithm that the frequency of it is no less than 75%, the algorithm performs well. Based on the above analysis, we can further optimize the space complexity of the algorithm.It is no difficult to prove that for the extended path$l_n = \{w_1 w_2 \dots w_n\}$, if we can guarantee that two n-1 sub paths are frequent at least, then the frequency of the algorithm can reach 75%. Duo to this, we improve the Fp-search algorithm with 0 road pruning.

The basic idea of Fp-search algorithm with 0 road pruning is that, for any frequent path$l_x = \{w_1 w_2 \dots w_x\}$, if$w_y$ is the previous node ofw_x, then $l = l_y + l_x$ will be expanded as the extended path. In this time, the frequency of pruning redundant paths is 50%. But only with a slight adjustment, can we improve the frequency to 75%. The basic idea of improvement is that, for any $l_x = \{w_1 w_2 \dots w_i w_x\}$ and $l_y = \{w_1 w_2 \dots w_i w_y\}$($w_x$ is a previous node ofw_y), if there is a path from w_x to w_y in the PE_Graph, we take the reverse search strategy to expand and add the path $l = \{w_1 w_2 \dots w_i w_y w_x\}$ to the extended path. Based on pruning of candidate 2-itemsets, let $f(l)$ present that the path l is frequent, then $S = \{w_t | l = \{w_1 w_2 \dots w_i\} \cap f(l + w_t)\}$ presents the frequent extended nodes of the path$l = \{w_1 w_2 \dots w_i\}$. For any $w_x, w_y \in S$(w_x is a previous node ofw_y), if there is a path from w_x to w_y in the PE_Graph, then $l + w_y w_x$ will be expanded as the extended path. The improved Fp-search algorithm with 0 road pruning enhances the ability to prune greatly. Compared to Fp-search algorithm with $k = 1,2,3$ road pruning, the performance of pruning is equivalent to the case of$k = 1$. But the improved Fp-search algorithm with 0 road pruning do not need to construct MP-tree, which reduces the space complexity greatly. In this paper, the improved Fp-search algorithm with 0 road pruning is called FPNMP-search algorithm.

```
**Algorithm** mining all frequent itemsets
Input:A transaction database DB and a minimum support $\xi$
Output:All frequent itemsets and the support count
Method:FPNMP-search (DB, $\xi$ )
1:Scan DB once. Collect the set of frequent items$L_i$and the
   support count.
2:Scan DB second, construct the PE_Graph gand Tid-tree t.
3:Gain tid-set of all node in t by a post- order
   traversal, t($i$)presents the tid-list of node i in t
4:for each node $\in [1, |L_1|]$
5: $freList \leftarrow [], l \leftarrow node$
6:for each $next \in childList(node)$
7:if $tid = t(node) \cap t(next) \geq \xi$
8:$freList \leftarrow next$
9:          $checklist = freList \cap childList(next)$
10:          $l \leftarrow next$
11:$search(next, checklist, tid)$
11:   end for
12:end for
13: [Function $search(node, checklist, tid)$]
14: for each$next \in childList(node)$(backward search
      checklist)
15:if$tid = t(node) \cap t(next) \geq \xi$
16:    $freList \leftarrow next$
17:       $checklist = freList \cap childList(next)$
18:        $l \leftarrow next$
19:    $search(next, checklist, tid)$
20:end for
```

The detailed procedures of FPNMP-search algorithm are shown below:

Step1: Initialize the PE_Graph and Tid-tree, and generate the tid-set of 2-itemsets

Step2: For any node $i \in L_i$, traverse the$childList(i)$. If the extended path is frequent with the traversal node *nod*e, then add *node* to $freList(i)$.$checkList(node)$is the intersection of $freList(i)$ and $childList(i)$. Then turn to step 3.

Step3: Traverse the$checkList(node)$. If the extended path is frequent with the traversal node *next*, then add *next* to $freList(node)$. $checkList(next)$ is the intersection of $freList(next)$ and $childList(next)$. Then turn to step 3. When the last node of $checkList(node)$ is extended, back to the cycle of step 2.

IV. Experiment and Analysis

The three databases (T40I10D100K, pumsb, kosarak) used in the experiment are true, downloaded from http://fimi.ua.ac.be/data/. The experimental platform used in this paperis Microsoft Windows 7 ultimate with Intel(R)Core(TM) i7-3370 3.40 GHz and 8G system memory. The detailed description of all datasets are shown as follows.

TABLE 1A SUMMARY OF DATASETS

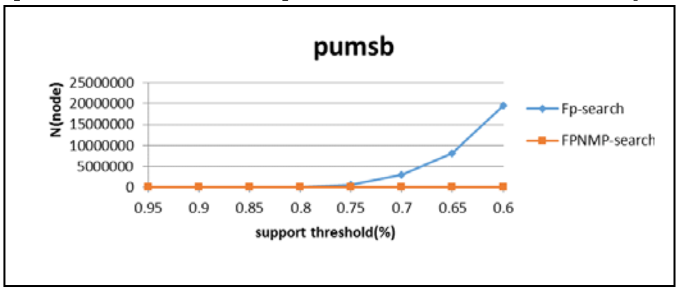
Database	Avg.length	#Items	#Trans
(a)T40I10D100K	40	1000	100000
(b)pumsb	74	7117	49046
(c)kosarak	8	41270	990002

In the table, Avg.length presents the average length of transactions. #Items presents the number of the items. #Trans presents the number of transactions in database DB. The T40I10D100Kand kosarak are sparse. The pumsb is density.

Based on the analysis of the algorithm, we can find that nodes in $freList$are the nodes in a frequent path, and this path is called local frequent path, which must exist in MP-tree. And all local frequent paths involved in all paths extension exist in MP-tree. Without constructing MP-tree, the space complexity of FPNMP-search is much less than Fp-search. The space complexity of two algorithm is not in an order of magnitude.

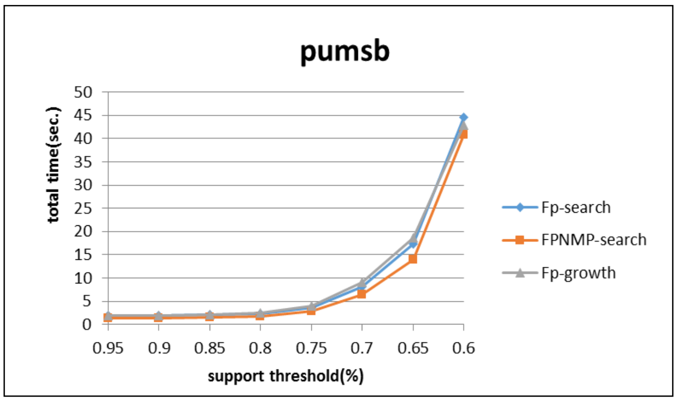
The result of experiment by using database pumsb is shown in Fig.1.

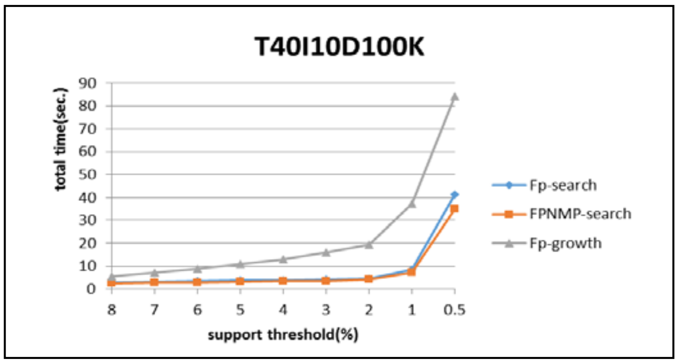
In Fig.2,we can find that the performance of FPNMP-search is better than Fp-search and Fp-growth, especially using sparse database. In the pumsb database, when $min\_sup =$



60%, originally, the total time of Fp-search is a little more than Fp-growth, while the FPNMP-search is still less than Fp-growth.

Fig.1 Comparison of FPNMP-search and Fp-search in space complexity





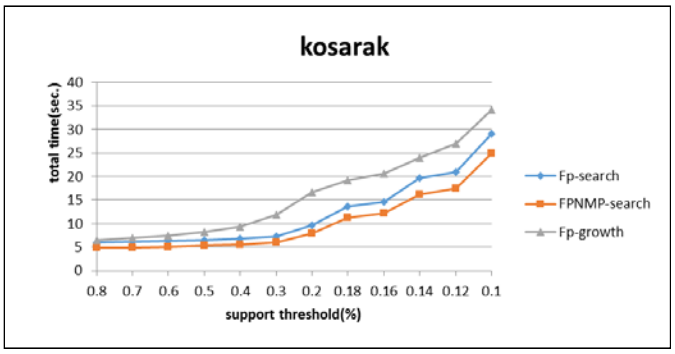


Fig.2Comparison of FPNMP-search, Fp-search and Fp-growth in total time

V. Conclusion

In this paper, we have deep research of frequent itemsets mining in association rules mining. In order to find more efficient algorithm for mining frequent itemsets, we propose an improved algorithm of Fp-search algorithm named FPNMP-search algorithm. Without reducing the ability to prune the redundant paths, the algorithm do not need to construct the MP-tree, which greatly reduces the space complexity of the algorithm and has a good performance in the time complexity. The results of the experiment show that FPNMP-search is more efficient than Fp-search and Fp-growth.

Acknowledgment

This work was supported by Collaborative Innovation Center of Wireless Communications Technology and the National Natural Science Foundation of China (Grant No. 6157060279).

References:

[1] Agrawal R, Srikant R. Fast algorithms for mining association rules. In VLDB’94, pp.487-499.

[2] J.S. Park, M.S. Chen,and P.S. Yu. An effective hash-based algorithm for mining association rules In SIGMOD’s95,pp. 175-186

[3] J. Han,J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. In Proceedings of ACM SIGMOD’00, pages 1-12, May 2000.

[4] M. Zaki and K. Gouda. Fast vertical mining using diffsets. In Proceedings of ACM SIGKDD’03, Washington,DC, Aug.2003.

[5] Hao JIANG,You_Jin LIAO and Shi-Meng NI A New Algorithm for Mining Frequent Itemset Using Efficient Data Structure International Conference on computer science and software engineering.2014

[6] Han J,Kamber M, Pei J. Data mining, Southeast Asia edition: Concepts and techniques[M]. Morgan kaufmann,2006.

[7] Ma C H P. Effective techniques for gene expression data mining[J]. 2006.

[8] Song Y. Efficient mining and maintenance of association rules in large datasets[D]. Concordia University, 2005.

[9] Won D, McLeod D. An efficient approach to categorising association rules[J]. International Journal of Data Mining, Modelling and Management, 2012, 4(4): 309-333.