

# A New Approach to Discover Periodic Frequent Patterns

Dr.K.Duraiswamy

K.S.Rangasamy College of Terchnology, Tiruchengode -637 209, Tamilnadu, India

E-mail: kduraiswamy@yahoo.co.in

B.Jayanthi (Corresponding Author)

Department of Computer Science, Kongu Arts and Science College

Erode – 638 107, Tamilnadu, India

E-mail: sjaihere@gmail.com

## Abstract

Mining Frequent Patterns in transaction database TD has been studied extensively in data mining research. However, most of the existing frequent pattern mining algorithm does not consider the time stamps associated with the transactions. Temporal periodicity of pattern appearance can be regarded as an important criterion for measuring the interestingness of frequent patterns in several applications. In this paper, we extend the existing frequent pattern mining framework to take into account the time stamp as periodicity i.e., the time stamp from the month January to June is as First Period and from July to December as Second Period, and discover frequent patterns for each period. An efficient tree based data structure called periodic-frequent pattern tree that captures the database TD in a highly compact manner and enables a pattern growth mining techniques to generate the complete set of Periodic-Frequent patterns. Example illustrating the proposed approach is given. The characteristics of the algorithm are discussed.

**Keywords:** Data mining, Frequent Patterns, Periodic-Frequent Patterns, Minimum Support

## 1. Introduction

A transaction database TD usually consists of a set of time-stamped transactions. Mining Frequent patterns or itemsets from a transaction database is one of the fundamental and essential operations in many data mining applications, such as discovering association rules, strong rules, correlations and many other important discovery tasks. The problem of mining frequent itemsets is formulated as finding all the itemsets that satisfy user specified support threshold. The important criterion for identifying the interestingness of the frequent patterns might be the shape of occurrence. i.e., whether they occur periodically, irregularly or mostly in specific time interval in the database.

In a retail market, among all frequently sold products, the user may be interested only on the periodically sold products. As for the stock market, the set of higher stocks indices that rise periodically may be of special interest to companies and individuals. We define such a frequent pattern that appears in a period/interval in a transaction database as a Periodic – Frequent patterns. In the previous work, most of the existing pattern mining algorithms do not consider the time stamps. In this paper, we extend the traditional frequent pattern mining framework to take into account the time-stamp i.e., in periods.

For example a transaction database TD has 16 transactions of 8 items. Let's focus on two patterns P1P2 and P1P3 without considering time information. P1,P2 and P1,P3 have the same significance in the traditional frequent pattern framework. Since they may have the same frequency of 62.50%. However interesting differences between these two patterns can be found after when we consider the time information. For simplicity consider one transaction per month. January to June Pattern P1,P2 occurs frequently and July to December pattern P1,P3 occurs frequently every month.

The above observation has shown that frequent patterns discovered by standard frequent pattern mining algorithm are not frequent for entire year. However such patterns are considered to be periodic patterns. The objective of the research presented in this paper is to distinguish such frequent patterns.

The rest of the paper is organized as follows. Section 2 gives the view of the related works. Section3 gives the statement of problem. Section4 presents the frequent pattern generation algorithm. Section5 gives the example of

the proposed algorithm. Section 6 shows the experimental results of the performance of the algorithm. Section 7 Concluding remarks are described.

## 2. Related Work

Since it was introduced in (R.Agrawal, T.Imielinski and A.N.Swami, 1993). The problem of frequent itemset mining has been studied extensively by many researchers. As a result, a large number of algorithms have been developed in order to efficiently solve the problem (R.Agrawal, R.Srikant, 1994, J.Han, J.Pel, Y.Yin, 2000). In practice, the number of frequent patterns generated from a dataset can often become excessively large, and most of them are useless or simply redundant. Thus there has been recent interest in discovering a class of new patterns, including maximal frequent itemsets (R.J.Bayardo, 1998, D.Burdick, M.Calimlim, J.Gehrke, 2001), Closed Frequent itemsets (J.Pei, J.Han, R.Mao, 2000, M.J.Zaki, C.Hsias, 2000). Temporal relationships among pattern occurrences were studied in (G.Tatavarty, R.Bhatnagar, B.Young, 2007). Periodic Pattern mining has also been studied as a wing of Sequential pattern mining (F.Maqbool, S.Bashir, A.R.Baig, 2006) in recent years.

The work presented here differs from the related work in some aspects as follows: Frequent Pattern tree (J.Han, J.Pel, Y.Yin, 2000) is generated for First and Second periods. Second mining of Frequent Pattern from the tree is done parallel for both periods.

## 3. Problem statement

The problem of mining association rules was introduced in (R.Agrawal, T.Imielinski and A.N.Swami, 1993). There are two steps in association rule mining. First step is to find Frequent itemsets and step is to generate Association rules. We focus on first step i.e., finding Frequent itemsets. Let  $I = \{i_1, i_2, i_3, \dots, i_m\}$  be a set of  $m$  items. A  $k$ -itemset is an itemset that contains  $k$  items. Let  $TD = \{T_1, T_2, T_3, \dots, T_n\}$  be a set of  $n$  transactions called a transaction database  $TD$ , where each transaction  $T_j$  ( $j \in \{1, 2, 3, \dots, n\}$ ) is a set of items such that  $T_j \subseteq I$ . Each transaction is associated with a unique identifier, called its TID. A transaction  $T_j$  contains an itemset  $X$  if and only if  $X \subseteq T_j$ . The Support Count of an itemset  $X$  is calculated as  $Sup_{TD}(X)/N$ , where  $Sup_{TD}(X)$  is the number of transactions in  $TD$  containing an itemset  $X$  and  $N$  is the total number transactions in the database.

The objective of periodic frequent pattern mining is to distinguish frequent patterns from different periods, that cannot be discovered through (R.Agrawal, T.Imielinski and A.N.Swami, 1993). In this work, an algorithm PFP-tree is proposed, to find the frequent patterns for different periods. More specifically, given a transaction database  $TD$ , a minimum Support and periods. i.e., the time-stamps converted into periods.

## 4. Proposed Algorithm

Algorithm PFP-tree

Input:

1. Transaction Database  $TD$  converted with periods
2. min support

Output:

Periodic-Frequent Pattern tree i.e., PFP-tree

1. Scan the  $TD$  once; generate a Frequent ( $F$ ) of 1-itemsets and their counts.
2. Generate an ordered frequency list ( $OL$ ) by filtering out infrequent items (items who do not pass the minimum support)
3. Sort the list ( $OL$ ) in frequency descending order as  $OL1$ . These ordered lists are used to build header tables.
4. Create the root of the PFP-tree  $T$  with label "Null"
5. For each transaction  $trans$  in  $TD$  do the following
6. Select and sort frequent items in  $trans$  according to  $OL1$ .
7. Let the sorted item list in  $trans$  be  $[p/P]$ , where  $p$  is the first element and  $P$  is the remaining list
8. Call  $Insert\_tree([p/P], T)$
9. End for

10. Function Insert-tree([p/P],T)
11. For Period  $i = 1$  to 2
12. If T has a child N such that  $N.itemName = P.itemName$
13. Then  $N.i.Count = N.i.Count + 1$
14. Else
15. Create New Node N with  $i.Count = 1$ , Parent linked to T
16. Node-link to the nodes with the same item-name via the node-link structure.
17. End if
18. If  $i = 1$  and  $P \neq \emptyset$
19. Then Insert\_tree(P,N)
20. else if  $i = 2$  and  $P \neq \emptyset$
21. Then Insert\_tree(P,N)
22.  $i = i + 1$ ;
- 23 End if
24. End for

### **Mining Frequent patterns:**

#### **Algorithm PFP Growth:**

Input: PFP-tree

Output: The complete set of Frequent patterns for each period.

Method: Call PFP-Growth(Tree, Null)

1. Procedure PFP-Growth(Tree, $\alpha$ )
2. If tree contains a single path P
3. Then for all combination 1-  $\beta$  and 2- $\beta$  of the nodes in the path P.( 1-  $\beta$  = period1 and 2- $\beta$  = period 2)
4. Generate pattern 1( $\beta U \alpha$ ) and 2( $\beta U \alpha$ ) with support = support of nodes in 1-  $\beta$  and 2- $\beta$ .
5. End for
6. Else
7. For all  $a_i$  in header table of tree do
8. Generate itemset 1-  $\beta = a_i U \alpha$  and 2- $\beta = a_i U \alpha$  with support =  $a_i.support$
9. Construct 1-  $\beta$  and 2- $\beta$  conditional pattern base and then conditional FPF-tree Tree 1-  $\beta$  and 2- $\beta$
10. If Tree 1-  $\beta \neq \emptyset$  and 2- $\beta \neq \emptyset$
11. Then call FPF-Growth(Tree,  $\beta$ )
12. End if
13. End for
14. End if

### **5. Example**

This Section shows the example to demonstrate the proposed algorithm to demonstrate the periodic frequent pattern mining

Table 1. Transaction Database TD

TID	List of Item ID's	Time Stamp
001	P1,P2,P3,P8	Jan 2006
002	P1,P2,P5	Feb 2006
003	P1,P2,P4	Mar 2006
004	P1,P2,P4,P5,P6	Apr 2006
005	P1,P2,P3,P4,P6	May 2006
006	P1,P4,P6	Jun 2006
007	P4,P5,P6	Jul 2006
008	P1,P2,P3,P4,P5,P6	Aug 2006
009	P1,P3,P4,P6	Sep 2006
010	P1,P3,P5	Oct 2006
011	P1,P2,P3,P6,P7	Nov 2006
012	P1,P3,P4,P5	Dec 2006

Phase I: Support count for Frequent-1 Itemset

Table 2. Frequent 1-Itemset Table (OL)

Item	Support Count
P1	11
P2	7
P3	7
P4	7
P5	5
P6	7
P7	1
P8	1

User specified minimum support count = 4, and prune the itemset that does not satisfy the minimum support count specified by the user. In the following table3 itemset P7 and P8 are pruned.

Table 3. Pruned Frequent 1-Itemset Table(OL1)

Item	Support Count
P1	11
P2	7
P3	7
P4	7
P6	7
P5	5

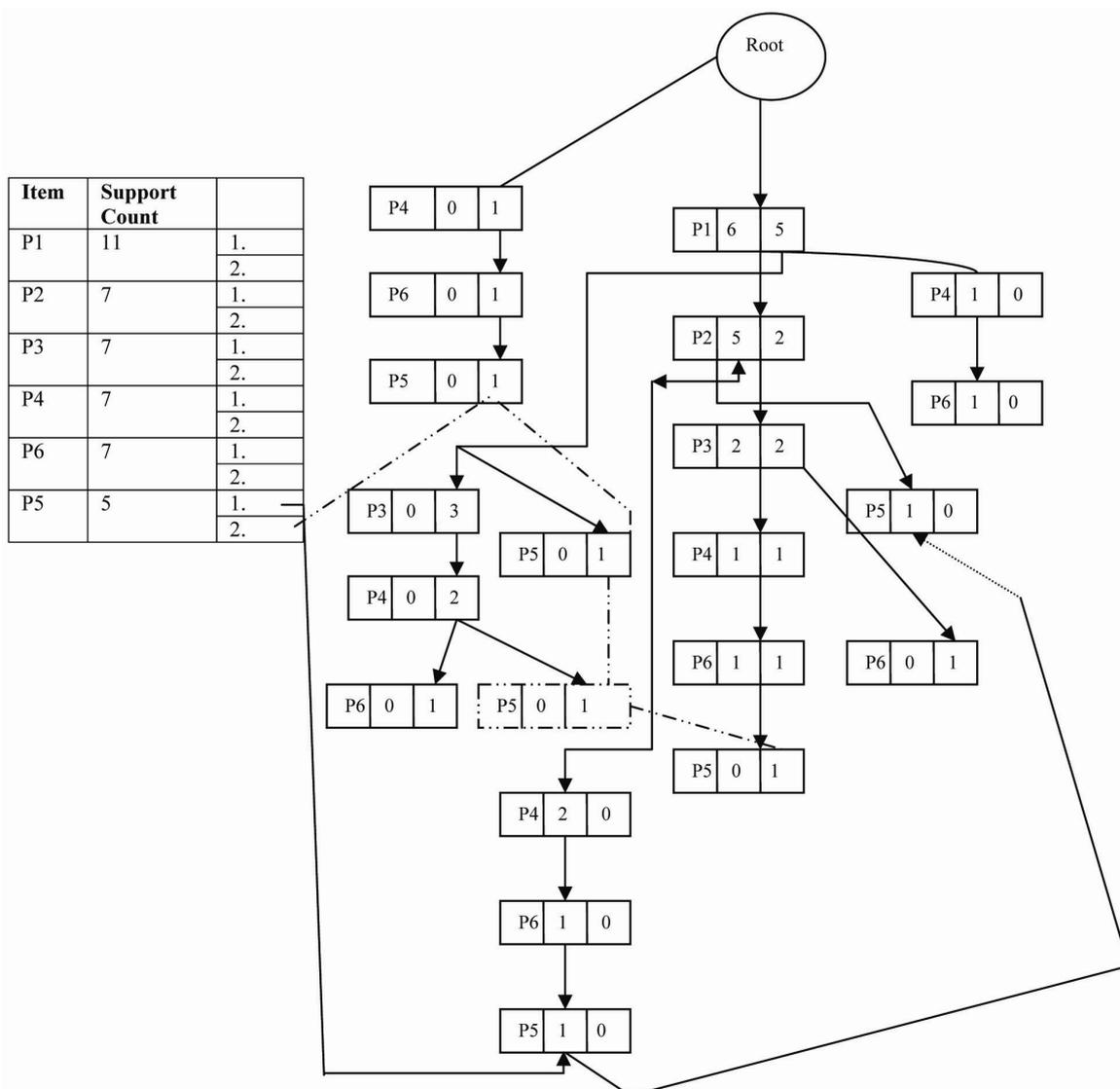
Phase 2: Intervals are assigned. The timestamp from Jan to June is considered to be period1 and from July to Dec are period 2.

Table 4. Transaction Database

TID	List of Item ID's	Time Stamp	Interval
001	P1,P2,P3	Jan 2006	1
002	P1,P2,P5	Feb 2006	1
003	P1,P2,P4	Mar 2006	1
004	P1,P2,P4,P6,P5	Apr 2006	1
005	P1,P2,P3,P4,P6	May 2006	1
006	P1,P4,P6	Jun 2006	1
007	P4,P6,P5	Jul 2006	2
008	P1,P2,P3,P4,P6,P5	Aug 2006	2
009	P1,P3,P4,P6	Sep 2006	2
010	P1,P3,P5	Oct 2006	2
011	P1,P2,P3,P6	Nov 2006	2
012	P1,P3,P4,P5	Dec 2006	2

Construct the FPF-tree using the proposed algorithm for period 1 and 2 in the same tree.

Header Table



Each node is divided into 3 parts. First part contains the item name and second parts contains support count for period 1 and third part contains support count for period 2. so that using the FPF-Growth mining techniques frequent itemset are mined parallel for both the periods. In Period 1 P5 has the P2,P1:1 and P6,P4,P2,P1:1 and for the period 2, P5 has the P6,P4,P3,P2,P1:1, P4,P3,P1:1, P6,P4:1 and P3,P1:1. Condition pattern base and condition pattern tree is constructed for P5 and finally frequent pattern for P5 is generated. Like wise it proceeds for the remaining items.

## 6. Analysis

This section analyses some of the characteristics of the proposed algorithm. The first characteristic is the time effect. Only twice database are scanned as well as frequent itemset for both periods are mined in parallel. A second characteristic is data structure for storing both the period is efficient.

## 7. Conclusion

In this work, frequent itemset is discovered for different periods in parallel and the algorithm for proposed work is presented. The proposed algorithm automatically generates the itemset. Example illustrating the proposed work is given and characteristics of the algorithm are analyzed.

## References

- Agrawal R, Imielinski T, Swami A. (1993). Mining association rules between sets of items in large databases. In Proc. Of the ACM SIGMOD Int. Conf. on Management of Data, Pages 207-216.
- Agrawal R, and Srikant R, (1994). Fast algorithms for mining association rules. In Proc. Of the 20<sup>th</sup> Int. Conf. on very Large Databases. pages 487-499.
- Bayardo R.J. (1998). Efficiently mining long patterns from databases. In Proc. Of the Int. ACM SIGMOD Conf., pages 85-93.
- Burdick .D, Calimlim, and Gehrke .J. (2001). Mafia: A maximal frequent itemset algorithm for transactional databases. In Proc. Of the 17<sup>th</sup> Int. Conf. on Data Engineering.
- Dong. G, Li .J. (1999). Efficient mining of Emerging Patterns: Discovering Trends and Differences. *Knowledge Discovery and Data Mining*, pages 43-52.
- Han .J, Pei .J, and Yin .Y. (2000). Mining Frequent patterns without candidate generation. In Proc. Of ACM-SIGMOD Int. Conf. on Management of Data, pages 1-12.
- Li .J, Ramamohanarao .K, Dong .G. (2000). Emerging patterns and Classification. In Proc. Of the 6<sup>th</sup> Asian Computing Science Conf. on Advances in Computing Science, Pages 15-32.
- Maqbool .F, Bashir .S, Baig >A.R. (2006). E-MAP: Efficiently mining asynchronous periodic Patterns. *Int.J. of Comp.Sc. and Net. Security* 6(8A), pages, 174-179.
- Tan .P, Kumar .V, and Srivastava .J. (2000). Indirect association: mining higher order dependencies in data. In Proc. Of the 4<sup>th</sup> European Conf. on Principles and Practice of Knowledge Discovery in Databases, pages 632-637.
- Tatavarty .G, Bhatnagar .R, young .B. (2007). Discovery of Temporal Dependencies between Frequent patterns in Multivariate Time Series. In. The 2007 IEEE Symposium on Computational Intelligence and Data Mining, pages, 688-696.
- Zaki .m .J, Hsiao.C. (2000). Charm: An efficient algorithm for closed itemset mining. In Proc. Of the 2<sup>nd</sup> SIAM Int. conf. on Data mining.