

AN IMPROVED FP ALGORITHM FOR ASSOCIATION RULE MINING

Rashmi Shikhariya*

Prof.Nitin Shukla*

Abstract—

Association rules among the large number of item sets is considered as an important aspect of data mining. Researchers developed a lot of algorithms and techniques for determining association rules. The main problem is the generation of candidate set. Among the existing techniques, the frequent pattern growth (FP-growth) method is the most efficient and scalable approach. It mines the frequent item set without candidate set generation. The main obstacle of FP growth is, it generates a massive number of conditional FP tree. In this research paper, we proposed a new and improved FP tree with a table and a new algorithm for mining association rules. This algorithm mines all possible frequent item set and also avoid the time consuming operation of searching whole tree and also improve time in add or delete transactions. It also provides the frequency of frequent items, which is used to estimate the desired association rules.

Keywords - *Associatio rule;FP-tree;Frequent pattern data;Frequency of the items.*

* Shri ram institute of technology, Jabalpur, India.

I. INTRODUCTION

Data mining is used to deal with large amounts of data which are stored in the database, to find out desired information and knowledge. Various data mining techniques such as, decision trees, association rules, and neural networks are already proposed and become the point of attention for several years. Association rule mining technique is the most effective data mining technique to discover hidden or desired pattern among the large amount of data. It is responsible to find correlation relationships among different data attributes in a large set of items in a database. Since its introduction, this method has gained a lot of attention. Association rules were first introduced in [1]. It provides information of the type of "if-then" statements. These rules are generated from the dataset and it derives from measurements of the support and confidence of each rule that can show the frequency of occurrence of a given rule. Association Analysis [1, 2, 4, 6] is the detection of hidden pattern or condition that occur frequently together in a given data. Association Rule mining techniques finds interesting associations and correlations among data set. An association rule [1,3,4,5] is a rule, which entails certain association relationships with objects or items, for example the interrelationship of the data item as whether they occur simultaneously with other data item and how often. These rules are computed from the data and, association rules are calculated with help of probability. It has a mentionable amount of practical applications, including classification, XML mining, spatial data analysis, and share market and recommendation systems. This rule measure with support to ensure every dataset treated equally in classical model. The perception of association rule mining suggests the support-confidence extent outline and condensed association rule mining to the discovery of frequent item sets. Rule support and confidence are two measures of interestingness. Association rules are regarded as appealing if a minimum support and a minimum confidence threshold is satisfied. This paper presents an efficient association rule mining technique with help of improved frequent pattern tree (FP-tree) and a mining frequent item set (MFI) algorithm. The main advantage of this mining is we can get all the frequent item set (1-item, 2-item and so on) with less affords. The outline of this paper is as follows: we discuss about some closely related work on association rule mining in Section II. Section III defines some contextual notations. At Section IV we described our key idea improved FP-tree with an example. Section V calculates the frequent items with MFI algorithm. Section VI shows the association rule mining and experimental results and Section VII concludes the paper.

II. RELATED WORKS

Algorithms for mining association rules from relational data have been implemented since long before. Association rule mining was first introduced at 1993 by R. Agrawal, T. Imielinski, and A. Swami [1]. After that many algorithms have been proposed and developed- Apriori [7], DHP [8], and FP-growth [9]. The Apriori algorithm [4] uses a bottom-up breadth-first approach to find the large item set. As it was proposed to grip the relational data this algorithm cannot be applied directly to mine complex data. Another well-known algorithm is FP growth algorithm. It adopts divide-and-conquer approach. First it computes the frequent items and characterizes the frequent items in a tree called frequent-pattern tree. This tree can also utilize as a compressed database. The association rule mining is performed on the compressed database with the help of this FP tree. This indicates that the dataset needs to be inspecting once. Also this algorithm does not require the candidate item set generation. So, in comparison Support S for a rule $X \Rightarrow Y$, denoted by $S(X \Rightarrow Y)$, is the ratio of the number of transactions in D that contain all the items in $X \cup Y$ to the total number of transactions in D [13]. Defined as

$$\sigma(X \cup Y)$$

with Apriori algorithm, it is much better in terms of efficiency [1]. But like other algorithms it also have certain

$$S(X \Rightarrow Y) = \frac{\sigma(X \cup Y)}{|D|} \quad (1)$$

|D|

disadvantages, it generates a large number of conditional FP trees. It generates this FP trees recursively as a procedure of mining. So the efficiency of the FP growth algorithm is not reasonable. But in proposed improved FP tree and MFI algorithm no need to generate conditional FP tree because the where the function σ of a set of items X indicates the number of transactions in D, which contains all the items in X.

Confidence C for a rule $X \Rightarrow Y$, denoted by $C(X \Rightarrow Y)$, is the ratio of the support count of $X \cup Y$ to that of the antecedent X.

$$C(X \Rightarrow Y) = \frac{\sigma(X \cup Y)}{\sigma(X)}$$

recursive element is separately stored in a different table. That reduces the existing bottleneck of the FP growth algorithm.

$$C(X \Rightarrow Y) =$$

(2)

$\frac{|X \cap Y|}{|X|}$

Many modified algorithm and technique has been proposed by different authors. Such as FP- tree and COFI based approach is proposed for multilevel association rules. Here except the FP- tree, a new type of tree called COFI- tree is proposed [11]. An Apriori based data mining technique is described at [12]. we use that example as the input of our proposed MFI algorithm and it is easily understandable that the new approach collect the association rule more efficiently. We will discuss about this in later sections.

III. CONTEXTUAL NOTIONS

In this section the basic concepts of association rule mining is illustrated. The problem of mining association rules can be explained as follows: There is the item set $I = \{i_1, i_2, \dots, i_n\}$ where I is a set of n discrete items, and consider D as a set of transactions, where each transaction, denoted by T , is a subset of items I . "Table 1" gives an example where a database D contains a set of transaction T , and each transaction consist of one or more items.

TABLE 1: SAMPLE TRANSACTION DATA.

Transaction	Items
T1	A,B
T2	A,C,D,E
T3	B,C,D,F
T4	A,B,C,D
T5	A,B,D,F

An association rule is an inference of the form $X \Rightarrow Y$, where $X, Y \subseteq I$ and $X \cap Y = \emptyset$. The set of items X is called antecedent and Y the consequent. Two properties support and confidence are generally considered in association rule mining. Other measures are also important but for the association rule mining, we will consider only the above two facts in this paper.

The minimum support S_{min} and minimum confidence C_{min} is defined by the user and the task of association rule mining is to mine from a data set D , that have support and confident greater than or equal to the user specified support value [14]. Association rule mining is a two-step process:

1. Find all frequent item sets: All the item set that occur at least as frequently as the user defined minimum support count.
2. Generate strong association rules: These rules must satisfy minimum support and minimum confidence and derived from the frequent item set.

For example, let assume the Minimum Support for the items at “table 1” is 40% and the minimum confidence is 60%. We want to determine the association rule $\{A, B\} \Rightarrow \{D\}$ is valid or not. So, support for this rule = $2/5 = 40\%$. (According to equation 1) and confidence = $2/3 = 66.66\%$. (According to equation 2)

This rule has support of 40% and confidence 66.6%. So; this is a valid association rule because it satisfies the Minimum support and confidence. The details procedure will be described in the following sections.

IV. IMPROVED FREQUENT PATTERN TREE (FP -TREE)

A. *Frequent Pattern Tree*

A Frequent pattern tree (FP-tree) is a prefix tree which permits the discovery of frequent item set without the candidate item set generation [5]. It is proposed to recover the weakness of some traditional data mining algorithm. For example: Apriori. It compresses a large data set into a structured and compact data structure, known as FP-Tree. FP –Trees follows the divide and conquers methodology. The root of the FP-tree is labeled as “NULL”. A set of item is defined as a child of the root. Traditionally a FP tree consists of three fields- Item name, node link and count. To help the tree traversal, a header table is used. It decomposes the data mining task in to smaller parts.

B. *Key Idea*

The main idea of this paper is, to construct an efficient FP- tree and mining frequent item set algorithm (MFI algorithm). This is because the FP-tree is highly condensed and surely it is smaller than the main database (for example, the transaction database as described as table 2) .It will reduce the expensive database scan in mining method. After forming a FP tree, the frequency of the FP-tree (F) is used as an input for MFI algorithm. It will generate the frequent item sets. These frequent item sets will be used to get strong association rules.

C. *Proposed Improved FP Tree*

Our proposed FP-Tree consists of mainly two elements- the tree and a table. The tree represents the correlation among the items more specifically and table is used to store the spare items. We called it spare table (Stable) which has two column- item_name and frequency. Item_name is the name of the items and frequency means how many times it occurs in Stable. The main reason to

introduce the spare table is, in traditional FP- tree a lot of branches are created and the same item appears in more than one node. But in our proposed FP-tree every distinct item has only one node. So it is simpler and efficient for further processing.

Definition 1: Let A is the set of items. Spare item S is defines as $S \in A$. There are two cases where we consider an item is spare.

Case 1: When the item has no single edge from the current root to its node. That means the item is already exists in the FP-tree. In proposed FP-tree, the root is not fixed, it changes. We will describe it shortly.

Case 2: The transaction items that do not contain the most frequent item. For example CD is the most frequent item in “table 2”.

D. Example of forming improved FP tree

For this purpose, we take the transaction data as follows.

TABLE 2: SAMPLE TRANSACTION DATA FOR THE FP TREE.

ID	Transactions
1	BOOKS,CD,VIDEO
2	CD, GAMES
3	CD,DVD
4	BOOKS,CD,GAMES
5	BOOKS,DVD
6	CD,DVD
7	BOOKS,DVD
8	BOOKS,CD,DVD,VIDEO
9	BOOKS,CD,DVD

And the user defined minimum support (S_{min}) is 2. Firstly, we scan the transactions and calculate the number of occurrence of each individual item (Table 3). We compare the number of occurrence with the S_{min} .

TABLE 3: TRANSACTION ITEMS AND NUMBER OF THEIR OCCURRENCE IN TRANSACTION

Item	Number of occurrence in transactions
CD	7
BOOKS	6
DVD	6
GAMES	2
VIDEO	2

As all the items satisfy the minimum support, so we have to consider all the items. According to the number of occurrence, all the transaction items are sorted in descending order (Table 4).

TABLE 4: DESCENDING ORDERED TRANSACTION DATA.

	Transaction items in descending order
1	CD,BOOKS,VIDEO
2	CD,GAMES
3	CD,DVD
4	CD,BOOKS,GAMES
5	BOOKS,DVD
6	CD,DVD
7	BOOKS,DVD
8	CD,BOOKS,DVD,VIDEO
9	CD,BOOKS,DVD

We divide the descending ordered transactions in two categories for better understanding of our proposed algorithm.

- i) Category 1: Transactions with most frequent item
(Table 5).
- ii) Category 2: Transactions without most frequent item (Table 6).

TABLE 5: TRANSACTION WITH MOST FREQUENT ITEM

ID	Transactions with most frequent item
1	CD,BOOKS,VIDEO
2	CD,GAMES
3	CD,DVD
4	CD,BOOKS,GAMES
6	CD,DVD
8	CD,BOOKS,DVD,VIDEO
9	CD,BOOKS,DVD

TABLE 6: TRANSACTION WITHOUT MOST FREQUENT ITEM

ID	Transactions without most frequent
5	BOOKS,DVD
7	BOOKS,DVD

The following algorithm, we use the concept as described at [16]

Algorithm 1: Construct improve FP-Tree

Input : Transaction database.

Output : Improved FP-tree, Frequency of each item in FP- tree, and the Stable count.

Step1 : Sort the items in the transaction database in descending order, according to the number of occurrence in transaction.

Step 2 : Create the root of the tree R.

Since it is a prefix tree, R= NULL.

Step 3 : For each transaction in transaction database **do**

Step 4 : Let a descending ordered transaction is represented by [p|Q].where p is the first item and Q is rest of the items in transaction.

Step 5 : If p= the most frequent item, do the following step 6 and step7.Otherwise go to step 8.

Step 6 : If the root R has a direct child node M, such that

M's item_name = p's item_name,

Increase the frequency of the item M, denoted as G.M by 1.

Transfer the root from R to p.

Step 7 : For each item in Q do the following steps up to Q is empty.

[a] Create a new node for each new item from root. Increase the frequency of the new item, G. new item by 1.

Transfer the root to new node.

[b] If an item, Q_i has no single edge from the current root to its node, Where Q_i is an item in Q and already exists in FP-tree. Then Q_i is a spare item.

Store Q_i with frequency 1 in spare table (Stable)

Step 8 : For each item in transaction [p|Q]

Store all items in Stable with frequency 1.

Step 9 : Calculate the total frequency of each item in stable, called Stable count.

Step 10: Output the FP-tree, frequency of each item in

FP-tree and total frequency of Stable count

Figure 1. Algorithm for proposed FP-tree.

In this proposal, we give our maximum attention on the most frequent item, because most of the relations are related with that particular item. So when we use the descending ordered transactions, then the most frequent item come first and the tree is formed based on that. The root of the tree is changed with the transactions (step 6 of algorithm 1). As the root is changing with the coming items, so, the correlation is more efficiently visible. Algorithm 1 compares items and classify them (step 5). This classification decides which is spare item (case 1 in definition 1). It enhances the efficiency of the FP- tree. After one node is created the other nodes will be created according to the frequency of occurrence of each item (step 7 of algorithm 1). If the same element occurs then the frequency of that item is increased by 1 (step 6 of algorithm 1).

Definition 2: The frequency of an item in the FP-tree is the number of time, it occurred in the FP- tree. It is presented as

{item_name: frequency of FP-tree}. for example in “figure 2”

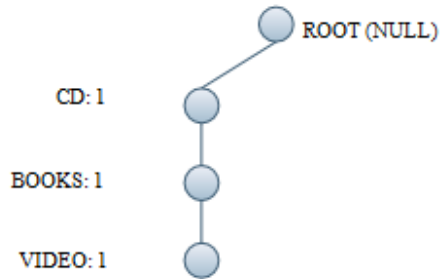
{BOOKS: 1}.

1) *First transaction: {CD, BOOKS, VIDEO}.*

The descending ordered list, in “table 4” is {CD, BOOKS, VIDEO}. The root of the improved FP-Tree is created and the value is “NULL”. As “CD” is the direct child node of the root, then it’s frequency is increased by 1 (step 6 in algorithm

1). Other items are in different nodes and the frequency of FP- tree is shown like {Item: FP-tree frequency} such as

{CD: 1}. The Stable is empty because the entire three items occurs for the first time and it contains the most frequent item CD. So, no spare item is valid here.



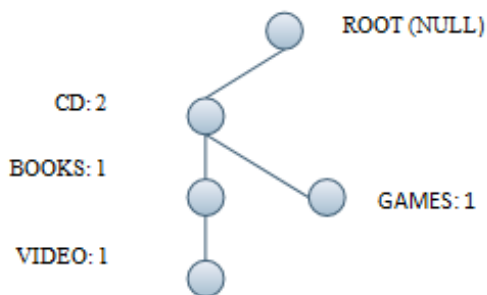
STABLE

Item_name	Freque

Figure 2. FP tree and Stable after first transaction.

2) *Second transaction: {CD, GAMES}*

There is a new item “GAMES”. And as described at step 6 of algorithm 1, that the root is changed. So, the new node will create from the node “CD” because this is the current root. That represents the correlation between two items better. It is notable that, frequency of the FP-tree for item “CD” has increased. The table is still empty because the items “CD” and “GAMES”, “CD” is already exists and “GAMES” a new item.



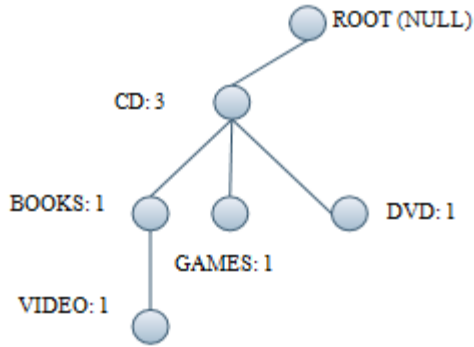
SPARE TABLE

Item name	Freque

Figure 3. FP tree and Stable after second transaction.

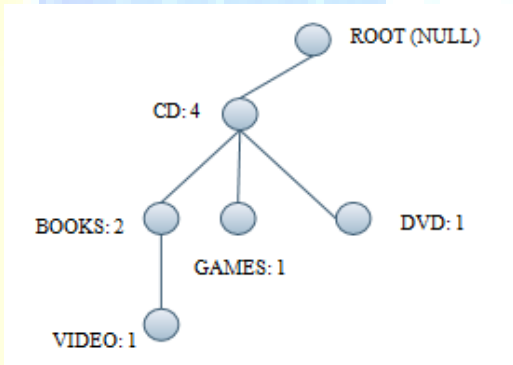
3) *Third transaction: {CD, DVD}*

Another new item “DVD” and the root is still “CD”, so tree is as below. As the item “CD” is the most frequent item in the transaction database, all the other items are generating with respect to it. The existing FP-tree keeps the track of the header to track of the first and next occurrence of an item, but in proposed approach, no track of items is taken. This eliminates the header table of existing FP-tree.



Item_name	Frequency
	SPARE TABLE

this improved FP-tree gives the more specific correlations among items and with Stable count, the association rule generation will be more error free because each elements are considered.



SPARE TABLE

Item_name	Freque
GAMES	1

Figure 5. FP tree and Stable after fourth transaction.

5) Fifth transaction: {BOOKS,DVD}

At fifth transaction, items are “BOOKS” and “DVD”. As this transaction is without most frequent item. This items will be stored in Stable. As described at step 5 and step 8 of algorithm1.

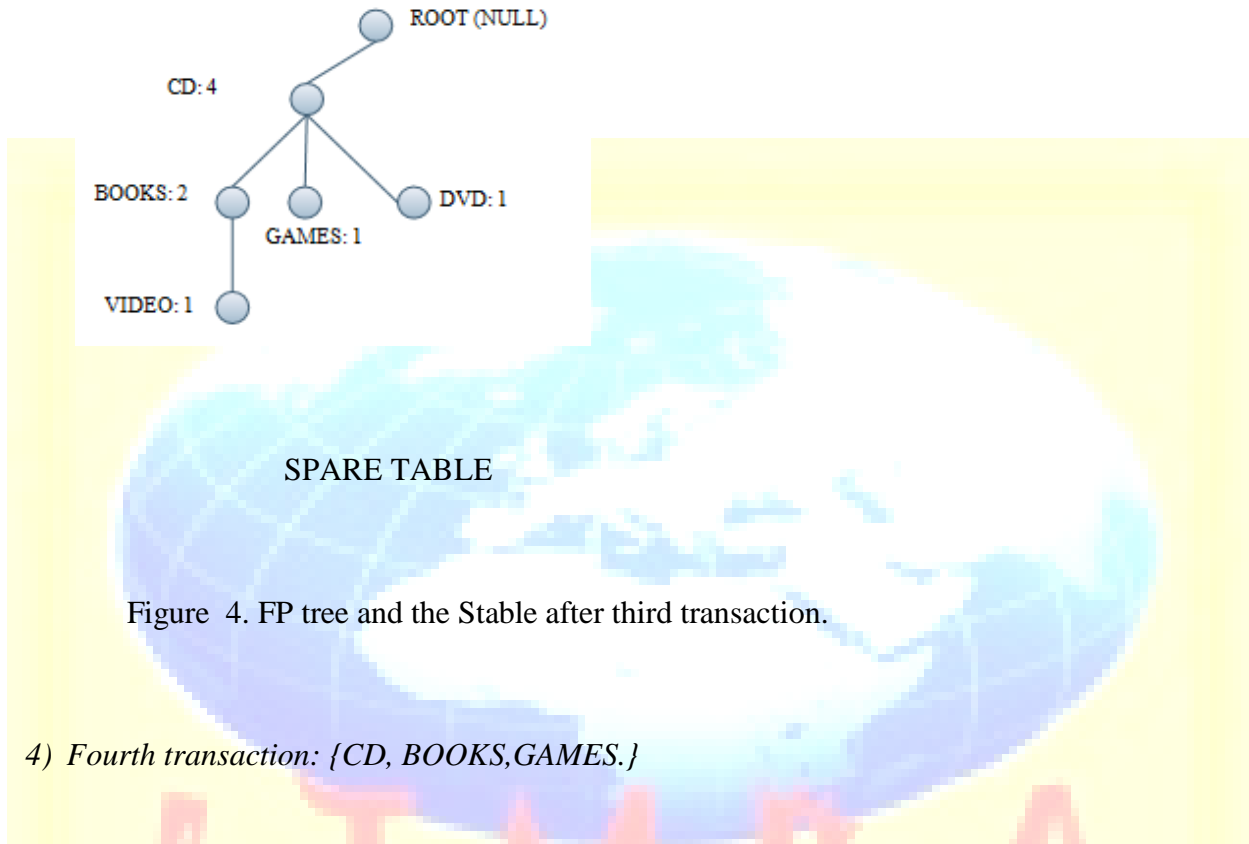


Figure 4. FP tree and the Stable after third transaction.

4) *Fourth transaction: {CD, BOOKS, GAMES.}*

Here, the node “CD” and “BOOKS” are in the same branch of the tree but the node “GAMES” is not. As the root is changing so, the current root is “BOOKS”. As there is no single edge from root “BOOKS” to node “GAMES”, the item is stored in Stable. As described at step 7 [b] of algorithm1 and “case 1” in definition 1. This indicates that the item “GAMES” occurs only one time with the most frequent item “CD” solely. So,

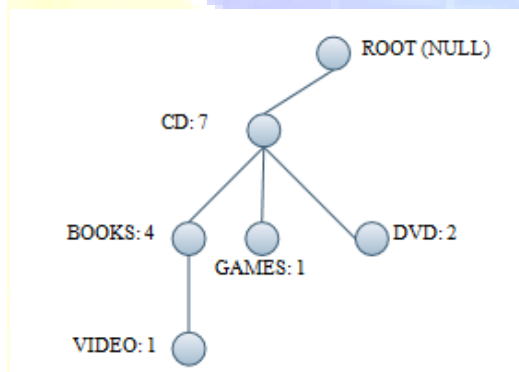
Figure 6. FP tree and Stable after fifth transaction.

Item_name	Frequency
GAMES	1
BOOKS	1
DVD	1

As we have total 9 transactions so, it will be enough if we show the specific parts of the FP-tree that is modified by our proposed algorithm.

6) Final transaction :{CD,BOOKS,DVD}

Finally the improved FP-tree is formed. In this tree, frequency of each item is different from the “number of occurrence of transaction” at “table 2”. Because this tree signifies, only the correlations among the items. But all the items are considered and operate with the help of Stable. The number of node in the tree is equal to the number of distinct items.



SPARE TABLE

Item name	Freque
GAMES	1
BOOKS	1
DVD	1
BOOKS	1
DVD	1
DVD	1
VIDEO	1
DVD	1

Figure 7. FP tree and Stable after all transactions. Now, from the step 9 of algorithm 1, Stable count is:

Item_name	Frequency
GAMES	1
BOOKS	2
DVD	4
VIDEO	1

Figure 8. Stable count after all transactions.

V. ALGORITHM 2: FOR MINING FREQUENT ITEM SET

(MFI algorithm)

To avoid candidate generation, the conventional FP growth algorithm needs to generate a large number of conditional pattern bases then calculate the conditional FP-tree. But in the case for large database it is inefficient and need a huge memory requirement that can over stack the main memory [1].we consider the MFI (mining frequent item set) algorithm to mine the frequent item set. The proposed algorithm uses the frequency of the proposed FP-tree and Stable count. The user determined support is used to calculate the frequency value of the frequent item sets.

The association rule mining have two steps- frequent item set generation and based on user defined support, find the valid association rules. The MFI algorithm will generate the frequent item set.

A. Mining Frequent Itemset Algorithm (MFI algorithm)

The MFI algorithm takes a simple approach to generate desired frequent item set that used to generate strong association rules. MFI takes the improved FP-Tree as an input with other parameters - user defined support S, Frequency in Stable count C, frequency of the FP-tree F, root R. This algorithm considers all possible combinations of items like existing FP-growth. As we have the user defined support, then based on the support and the frequency of FP-tree, all three possible cases are considered. That is, the support can be equal, greater or less than the FP-tree frequency. The MFI calculates the frequent item set and their corresponding frequency from this principle.

Definition 3: Frequency of the frequent item set is defined by the MFI algorithm, which is calculated with respect to user defined support count. One thing need to be specified there, the FP-tree frequency (F) and the frequent item set frequency (FF) is different. The FP -tree frequency can consider here as an input for the frequent item set frequency.

IF the frequency of the item in FP-tree is equal to the user defined support, then the frequency of frequent item sets for that item is FP-tree frequency. And the frequent items should be only those items, which have greater or equal FP-tree frequency at FP-tree. (As described at line 3-5 in algorithm2). Because association rules are only concerns in those items that satisfy the minimum support.

In the case where support is greater than the FP- tree frequency, the frequency for the frequent item sets is the total of the FP- tree frequency and the Stable count (line 6, 7). The main reason for adding is, as the FP tree represent the most frequent items, then some element (spare items)

will store in Stable though they have high frequency of occurrence in main transaction dataset. The frequent items should be all intermediate nodes of the path up to the most frequent item node (line 8) , as all the nodes at same path represent relation between nodes and for strong association rule, all possible relations need to be counted.

In the last case where support is less than the FP- tree frequency, the frequency value for frequent item set is the frequency of the FP-tree (line 9). As the support is less than the frequency, no need to add the frequency of the Stable count. The frequent item sets are the only immediate parent nodes of the FP-tree (line 10). Because all the parent node in proposed FP-tree have greater frequency than the child node.

Algorithm 2 For Frequent Item set

```

1: Mining Frequent Item set (FP-tree, S, C, F, R)
2: for each item i in FP-tree where (i! = R) do
3:   if i.S = i.F then \* i.S is the support of the item
      And i.F is frequency of the item in FP tree */
4:     frequency of the frequent item set, K = i.F
5:     Generate item set, P = (i  $\square$   $\mu$ ) with the
      frequency value of the tree.
      /*  $\mu$  = All possible combinations of the item and
      nodes with higher frequency in FP-tree */ Frequent
      item set is written in {P: K} format.
6:   else if i.S > i.F then
7:     Frequency of frequent item set K = i.F + C
      /* C=frequency in Stable count*/
8:     Generate item set, P = (i  $\square$   $\alpha$ )
      /* $\alpha$  = All possible combination of item and
      all intermediate nodes up to most frequent item
      node in FP-tree*/
9:   else Frequency of the frequent item set K = i.F
10:    Generate item set, P = (i  $\square$   $\beta$ )
      /* $\beta$  = All possible combinations of item
      and its parent node in FP-tree*/
11: end of for

```

Figure 9. Algorithm for mining frequent item set.

B. Experimental Results of MFI Algorithm

Consider the transaction data at “table2”.The MFI algorithm, derives the following frequent item set and the respective frequency of the frequent item sets.

TABLE 4: FREQUENT ITEM SET

Item	Frequent Item Set
CD	{CD:7}
BOOKS	{BOOK:4}; {BOOK,CD:4}
VIDEO	{VIDEO:2};{VIDEO,BOOK:2};{VIDEO,CD:2}; {VIDEO,BOOK,CD:2}

GAM	{GAMES :2};

Figure 10. Stable count after all transactions.

The procedure to get the frequent item set is easily understandable. For CD, The frequency according to the FP tree is “7” and the user defined support threshold is 2. So, according to our algorithm, if FP-tree frequency is greater than support then:

The frequent item set frequency is 7 (line 9 of the MFI algorithm). And for frequent item sets, consider only the parent node as given at line 10. So, the frequent item set is {CD: 7}.

In the case of VIDEO, the frequency at FP tree is 1. That is less than support. So,

i) The frequency is = frequency at FP tree + Frequency at
Stable count (line 7 of MFI).

So, the frequency is $1+1=2$; and the all intermediate nodes up to most frequent item node "CD" are BOOKS and CD. So the frequent item sets are

{VIDEO:2}; {BOOK, VIDEO:2}; {CD, VIDEO:2}; {BOOK, VIDEO, CD : 2}.

In the case of DVD, the frequency of FP-Tree is 2 that is equal to support. According to the line 3 of the MFI algorithm, frequency of the frequent item set is also 2. for frequent item set only those item whose have higher FP-Tree frequency that is- CD and BOOKS (line 5). So, the frequent item sets are

{DVD : 2}; {DVD, BOOKS : 2}; {VIDEO, CD:2};
{VIDEO, BOOK, CD:2}.

Now, it is notable that in every previous approach to find the frequent item set generating like [2] generates the 1-itemset,

2-itemset and so on. But in our approach all types of possible item set which satisfy the minimum user defined support is generated once. This is make our algorithm efficient than the others. Now it is quite easy to define the association rules from the frequent item set.

VI. ASSOCIATION RULE MINING

Let minimum confidence threshold is 60%. The resulting association rules are shown below. For this procedure we will use the term “FF” for frequent item set frequency.

R1: BOOKS and VIDEO \Rightarrow CD

Confidence= $\frac{FF \{BOOKS,CD,VIDEO\}}{FF \{BOOKS,VIDEO\}}$

= $\frac{2}{2} = 100\%$ and R1 is selected.

R2: VIDEO and CD \Rightarrow BOOKS

Confidence= $\frac{FF \{BOOKS, VIDEO, CD\}}{FF \{VIDEO, CD\}}$

= $\frac{2}{2} = 100\%$ and R2 is selected.

R3: BOOKS and CD \Rightarrow VIDEO

Confidence= $\frac{FF \{BOOKS,CD, VIDEO\}}{FF \{BOOKS, CD\}}$

= $\frac{2}{4} = 50\%$ and R3 is rejected.

R4: BOOKS \Rightarrow VIDEO and CD

Confidence= $\frac{FF \{BOOKS, VIDEO, CD\}}{FF \{BOOKS\}}$

= $\frac{2}{4} = 50\%$ and R4 is rejected.

VII. CONCLUSION AND FUTURE WORKS

In this paper, we proposed a new improved FP tree and a corresponding algorithm MFI to mine association rules. Our approach is surely efficient than the existing FP-growth based procedures.

It reduces the total step of the procedure and also takes less memory. In future, we will develop the association rule mining tool for XML data mining.

ACKNOWLEDGMENT

R5: VIDEO \Rightarrow BOOKS and CD

This research was supported by Basic Science Research

Confidence = $FF \{BOOKS, VIDEO, CD\} / FF \{VIDEO\}$

$= 2/2 = 100\%$ and R5 is selected.

R6: CD \Rightarrow BOOKS and VIDEO

Confidence = $FF \{BOOKS, VIDEO, CD\} / FF \{CD\}$

$= 2/7 = 28\%$ and R6 is rejected.

In this way, we have found three strong association rules. As we described earlier, we are comparing the example given at [2]. For experiment, we consider the data from "table 2". The figure 11 shows the comparisons. We compare with respect to Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2010-0013487).

REFERENCES

- [1] R. Agrawal, T. Imielinski, and A. Swami.. “Mining association rules between sets of items in largedatabases”. In Proceedings of the 1993 ACM SIGMODInternational Conference on Management ofData, pages 207-216, Washington, DC, May 26-281993.
- [2] R Srikant, Qouc Vu and R Agrawal. “Mining Association Rules with Item Constrains”. IBM Research Centre, San Jose, CA 95120, USA.
- [3] Ashok Savasere, E. Omiecinski and ShamkantNavathe“An Efficient Algorithm for MiningAssociation Rules in Large Databases”. Proceedingsof the 21st VLDB conference Zurich, Swizerland,1995.four important procedures, plotted in X-axis. They are-
- [4] R. Agrawaland R, Shrikanth, “Fast Algorithm forMining Association Rules”. Proceedings OfVLDBconference, pp 487 – 449, Santigo, Chile, database scan, conditional pattern base, conditional FP-tree generated and candidate set generated by three algorithms-1994.
- [5] Arun K Pujai “Data Mining techniques”. UniversityPress (India) Pvt. Apriori, FP-growth and proposed MFI algorithm. Y-axis Ltd. 2001
- [6] J. Han and M. Kamber.Data Mining: Concepts and Techniques. Morgan represents the number of times the four procedures occurs. Kaufman, San Francisco, CA,2001.
- [7] J. S. Park, M.-S. Chen, and P. S. Yu, “An effective Hash-Based Algorithm for Mining Association Rules”,Proceedings of the ACM SIGMOD, San Jose, CA, May1995, pp. 175-186.