A Novel Algorithm for Cross Level Frequent Pattern Mining in Multidatasets

B. Jayanthi
Associate Professor in MCA
Kongu Arts and Science College
Erode, Tamilnadu, India

Dr. K. Duraiswamy
Dean
K.S.Rangasamy College of Technology
Tiruchengode, Tamilnadu, India

ABSTRACT

Frequent pattern mining has become one of the most popular data mining approaches for the analysis of purchasing patterns. There are techniques such as Apriori and FP-Growth, which were typically restricted to a single concept level. We extend our research to discover cross - level frequent patterns in multi-level environments. Unfortunately, little research has been paid to this research area. Mining cross - level frequent pattern may lead to the discovery of mining patterns at different levels of hierarchy. In this study a transaction reduction technique with FP-tree based bottom up approach is used for mining cross-level pattern. This method is using the concept of reduced support.

Keywords: Data mining, cross – level frequent Patterns, FP-tree.

1. INTRODUCTION

Association rule mining is as important research subject put forward by Agrawal in reference [1]. Association rule mining techniques can be used to discover unknown or hidden correlation items found in the database of transactions. The pattern of mining association rule could be decomposed into two sub problems. First, mining of frequent itemsets/patterns and generating association rule from frequent patterns is next [1] [3]. Finding frequent patterns becomes the main work of mining association rules [2]. However, previous work has been focused on mining association rules at a single concept level as well as multiple levels. There are applications which need to find "cross level" association rule at multiple concept levels. For example, besides finding 80% of customers that purchase milk may also purchase Bread, it could be informative to also show that 75 % of people buy Wheat bread if they buy 2% milk or 70% of people buy milk if they buy wheat bread. The association relationship in the later statement is expressed at a lower level but carries more specific and concrete information than that in the former. This requires progressively deepening the knowledge mining process for finding refined knowledge from data. The necessity for mining multiple-level association rules or using taxonomy information at mining association rules has been observed by other researchers [2, 4, 11, and 12].

One approach to multilevel mining would be to directly exploit the standard algorithms in this area – Apriori [1] and FP-Growth [3] by iteratively applying them in a level by level manner to each concept level. In this paper, we introduce a new study in discovery of frequent patterns based on the FP-tree [5]. Our approach is different from FP-Growth algorithm [3] which needs to recursively generate conditional FP-trees such that a large amount of memory space needs to be used.

Our approach minimizes I/O costs by applying transaction reduction technique and applying the resulted transactions in FP-tree as input to subsequent iterations of the mining process. Our method adopts a bottom-up approach, with a leaf to root traversal, so as to identify frequent patterns existing between arbitrary classification levels. Our method reduces the search spaces without losing any patterns.

This paper looks to mine cross – level frequent patterns from mulitdatasets and proposes a continuation and extension work of the work in [6]. The paper is organized as follows section 2 discusses related work. The basics behind multilevel association rules are given in section 3. We present the description about transaction reduction concept and Bottom-up cross-level frequent pattern mining in section 4. Experiments and results are presented in section 5. Section 6 concludes the paper and our future work.

2. RELATED WORK

Since association rule mining was introduced in [1]. The problem of mining frequent pattern 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 [2,3]. In practice the number of work has been focused on mining association rules at single concept level. Thus there has been recent interest in discovering multiple level association rules. A new approach to mine frequent patterns for multidatasets has to be considered. Work has been done in adopting approaches originally made for single level datasets into techniques usable on multilevel datasets. The paper in [7] shows one of the earliest approaches proposed to find frequent patterns in multilevel datasets and later revisited in [4]. This work primarily focused on finding frequent pattern at each level in the dataset. The paper in [8] proposed a novel method to extract multilevel association rule based on different hierarchical levels by organizing and extracting frequent patterns. One adaptation of Apriori to multilevel datasets is a top-down progressive deepening method by Thakur, Jain & Pardasani in [9]. This approach was developed to find level – crossing association rules by extending existing multilevel mining techniques and uses reduced support and refinement of the transaction table at every hierarchy level .Due to the refinement of the transaction table some cross-level patterns were missed by this method and it scans multiple times the table to generate cross-level patterns. T.Eavis proposed an algorithm in paper [5] to mine cross-level frequent pattern by multiple FP-trees to generate cross-level pattern. The paper in [10] has proposed a new algorithm for transaction reduction based frequent pattern mining in single concept level.

However, the majority of work has proposed on finding frequent patterns as efficiently as possible, but it relies on using standard algorithms. The previous work in [6] has implemented to find large 1 frequent pattern for all levels using new method CCB-tree. In this work, we attempt to reduce the unwanted patterns and transactions using transaction reduction technique and applying the resulted transactions in FP-tree as input to subsequent iterations of the mining process. Our method adopts a bottom-up approach, with a leaf to root traversal with single FP-tree generation, so as to identify frequent patterns existing between arbitrary classification levels. Our method reduces the I/O costs and search spaces without losing any patterns.

3. MULTILEVEL ASSOCIATION RULE

We assume that the database contain 1) an item dataset which contain the description of each item in I in the form of (Ai, description), where Ai € I and 2) a transaction dataset, T, which consist of a set of transaction (Ti { Ap,…. Aq,}), where Ti is a transaction identifier and Ai € I for (for I = p….q).

To find relatively frequent occurring patterns and reasonably strong rule implications, a user or an expert may specify two thresholds: minimum support, σ' and minimum confidence, φ. For finding multiple-level association rule, different minimum support and/or minimum confidence can be specified at different levels.

Definition 1: The support of an item A in a set S, σ(A/S), is the number of transactions(in S) which contain A versus the total number of Transactions in S.

Definition 2: The confidence of A→B in S, φ(A→B/S), is the ratio of σ(AUB/S) versus σ(A/S), i.e., the probability that item B occurs in S when item A occurs in S.

The definition implies a filtering process which confines the pattern to be examined at lower level to be only those with large support at their corresponding high level. Based on this definition, the idea of mining multiple- level association rules is illustrated below.

Table1: A sales transaction table

transaction_id	Bar_code_set
351428	{17325, 92108, 55349…}
982510	{92458, 77451, 60395…}
----	----

Example 1: Let the query to be to find multiple-level association rule in the database in Table 1 for the purchase patterns related to Category, Content and Brand of the food which can only be stored for less than three weeks.

Table 2: A sales_item (description) relation

Bar_code	Category	Brand	Content	Size	Storage_pd	price
17325	Milk	Foremost	2%	1(ga)	14(days)	$3.89
----	----	----	---	----	----	----

Table 3: A generalized sales_item description table

GID	Bar_Code_Set	Category	Content	Brand
112	{17325, 31414, 91265}	Milk	2%	Foremost
----	----	----	---	----

The relevant part of the sales item description relation in Table 2 is fetched and generalized into a generalized Sales_item description table, as shown in Table 3, in which is tuple represent a generalized item which is the merge of a group of a tuples which share the same values in the interested attributes. For example, the tuple with the same category, content and brand in Table 2 are merged into one, with their bar codes replace by a bar-code set. Each group is then treated as an atomic item in the generation of lowest level association rules. For example, the association rule generated regarding to milk will be only in relevance to (at the low concept levels) brand (such as Dairyland) and Content (such as 2%) but not to size, producer, etc.

The taxonomy information is provided in table 3. Let Category (such as "milk") represent the first-level concept, content (such as "2%") for the second level one and brand (such as "Foremost") for the third level one. The table implies a concept tree like Fig.1.

The process of mining Multiple-level association rules is actually will be starting from top-most concept level. Let the minimum support at this level be 5% and the minimum confidence is 50%. One may fine the Large 1-itemset: "bread (25%), meat (10%), and milk (20%), Vegetable (30%).

At the second level, only the transactions which contain the large items at the first level are examined. Let the minimum support at this level be 2% and the minimum confidence is 40%. One may find frequent 1-itemsets: "lettuce (10%), Wheat bread (15%), white bread (10%, 2% milk (10%)..." The process repeats at even lower concept level until no large patterns can be found.

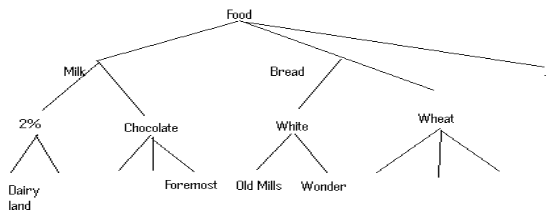

Fig. 1: Taxonomy for the relevant data items.

4. MINING FREQUENT PATTERNS

From the beginning of association rule mining in [1], the first step has always been to find the frequent patterns or itemsets. Here in this section we first introduce transaction reduction technique in multilevel datasets and then our work to mining cross-level frequent pattern using FP-tree based bottom up method.

4.1.Proposed Transaction Reduction Technique

This algorithm is based on reducing non candidate patterns and transactions. The idea is based on the theorem described in next subsection.

4.2.Theorem and proof

Theorem: If c € Fk and c.support < min.support, Titems ≤ k, k = 1, then c is useless in Fk+1 where Fk is Frequent pattern, c is an itemset in each transaction and Titems is total item count in each transactions.

Proof 1: [For c € Fk]

Consider a transaction Ti = {T1, T2, T3… Tm}. Let T1 = {a1, a2, a3… an}, T2 = {a1}. Since c is a hierarchy data. Whenever the lower-level items of c achieve a support count, the higher –level items should be added into the reduced transaction table. For Example data c is 111, 211... Satisfies support count, therefore the higher –level items of 11*, 21*… and the next higher-level also 1**, 2** should be added into the reduced transaction table. If lower-level items of c does not satisfy the min.support, then lower-level items of c is removed from the reduced transaction table. Hence the proof.

Proof 2: [For Titems ≤ k where k = 1]

Now consider the same transaction Ti = {T1, T2, T3… Tm}. Let T1 = {a1, a2, a3… an}, T2 = {a1}. During frequent k+1 pattern generation transaction T2 requires at least 2 as item count and if not then, Ti can be rejected from the transaction table. Hence the proof.

Let us consider the following Example with sample database.

Table4: Sample Database

TID	Items
T1	**{111, 121, 211, 221}**
T2	**{111, 211, 222, 323}**
T3	**{112, 122, 221, 411}**
T4	**{111, 121}**
T5	**{111, 122, 211, 221, 413}**
T6	**{113, 323, 524}**
T7	**{131, 231}**
T8	**{323, 411, 524, 713}**

CCB – Tree Algorithm [6] has been used to find multilevel frequent 1 pattern.

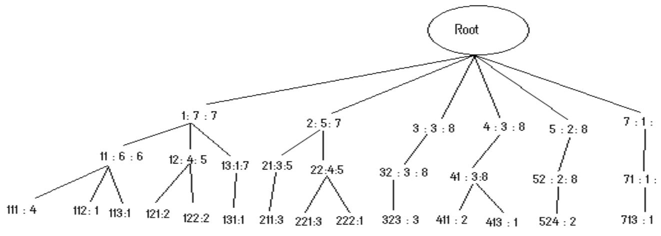

Fig. 2: CCB-tree

CCB-Tree Mining Process:

Minimum support for all levels is 4, 3, and 3:

Mining starts from the left most initial node i.e., from 1**: 7 > min.support and its descendents 11*:6>3 and 111>3. But 112,113<3 so it's considered to be a large 1 frequent pattern.

Finally frequent 1 pattern for level 1: 1**, 2** Level 2: 11*, 12*, 21*, 22* Level 3:111,211,221.

Table5: Reduced Transaction Table - TRD

TID	Items
T1	**{111, 12*, 211, 221}**
T2	**{111, 211, 22*}**
T3	**{11*, 12*, 221}**
T4	**{111, 12*}**
T5	**{111, 12*, 211, 221}**
T7	**{1**, 2**}**

By the proposed transaction reduction technique T6 contain 11* i.e., 1 item count so it is removed from the table4 and reduced transaction table is produced.

4.3.FP-tree Generation

In this section we introduce and describe our approach in details and given a running example to illustrate our algorithm for mining cross-level frequent pattern by traversing a FP-tree. We employ the frequent pattern tree structure to compress a larger database into a highly condensed much small data structure which avoids costly, repeated database scans. If two access data share a common prefix according to some sorted order of frequent patterns, the shared parts can be merged using one prefix structures as long as the count is registered properly. If the frequent items are sorted in descending order of their frequency, there is better chance that more prefix strings can be shared.

Algorithm 1 (FP-tree construction)

Input: Reduced Transaction Table – TRD

Output: Its frequent pattern tree, FP-tree

Method: The FP-tree is constructed in the following steps.

1. Create the root of an FP-tree, T, and label it as null. For each access data in reduced transaction table do the following.

1.1 Sort the frequent patterns F in the reduced transaction table with their support (lower- level to higher level) in descending order. i.e. F is a hierarchy data.

1.2 Let the sorted frequent pattern list in TRD be [p│P], where p is the first element and P is the remaining list. Call insert_tree ([p│P],T).

1.3 Function insert_tree ([p│P],T) is performed as follows. If T has a child N so that N.item-name = p.item-name, then increment N's count by 1; otherwise create a new node N and let its count be 1, its parent link be linked to T, and its node-link be linked to the nodes with the same item-name. If P is not empty, call inset_tree(P,N) recursively.

Table6: Reduced Transaction Table – TRD with sorted items

TID	Items
T1	**{111, 211, 221, 12*}**
T2	**{111, 211, 22*}**
T3	**{ 221, 11*, 12*}**
T4	**{111, 12*}**
T5	**{111, 211, 221, 12*}**
T7	**{1**, 2**}**

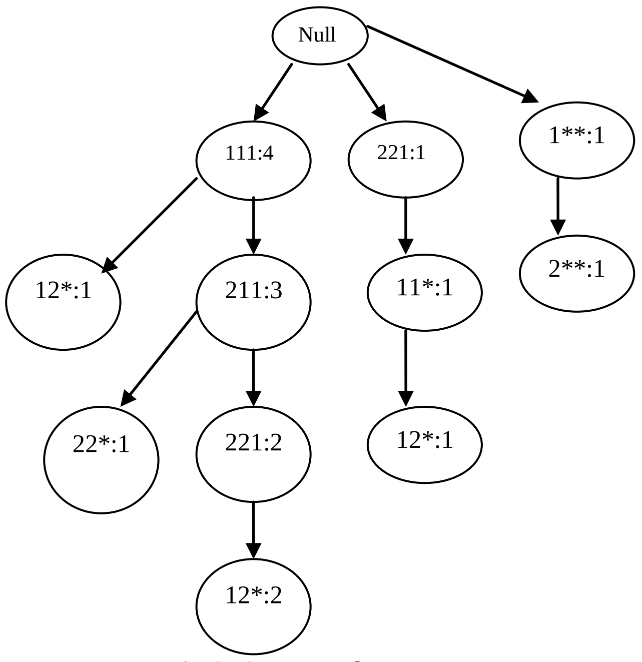

Fig. 3: Cross-Level FP-tree

4.4.Frequent Pattern Generation

After generating a FP-tree, the next phase is to generate candidate itemsets and find frequent patterns. Cross-level frequent pattern with bottom up approach starts from the leaf nodes of an existing FP-tree and traverses each branch upwards until it reaches its root. We begin by scanning the tree and identifying its leaf nodes. A pointer to each leaf is then inserting into the leaf node array. We now perform a bottom up scan of each leaf node until we reach the root. Meanwhile each node visited is conserved into temporary buffer for recording the passing path when a node with support count is visited. Candidate Generation keeps the path from starting node i.e. leaf node to the current node and generate all combinations of candidate 2-itemset. Thus when it comes to generate cross-level itemsets, we use a single global cross-level threshold. Only items from all levels that are above this threshold can be considered as frequent. The candidate itemset which satisfies the minimum support count that candidate can be used for next level processing, the node which does not satisfy minimum .support can be ignored and candidate generation does nothing for this. After finding frequent 2-itemsets from all sub trees. Next traversal is Candidate generation for frequent 3 itemsets. The supports for all the candidate k-itemsets ($k \geq 3$) can be computed and the frequent k-itemset can be obtained. This process proceeds until to find frequent k patterns.

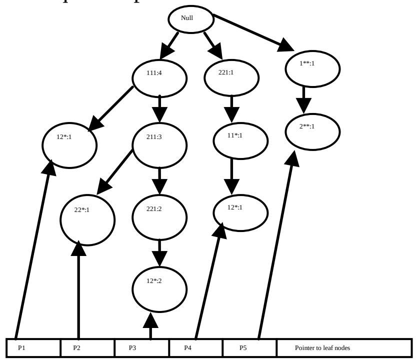

Fig. 4: Cross-Level FP-tree with pointers to the Leaf node.

For example, from Fig. 4 we start from the first pointer P1, candidate 2 itemsets are 12*, 111:1, 12*, 11*:1, 12*, 1**:1. Since 1** is the ancestor of 12* so 12* , 1** is eliminated. For the next pointer, (22*,211,111:1) i.e., 22*,211: 1, 22*,21*:1, 22*,111 :1, 22*,11*:1,22*,1**:1, 211,111:1, 211,11*:1, 211*1**:1, 2**,1**:1.

Algorithm2 - MCLFPT (Frequent Pattern Generation)
Input: FP-tree, minimum support count.
Output: Frequent 1 to k pattern.
Method: Bottom up mining method with candidate generation process

1. Find the leaf node for all sub trees and insert a pointer to each leaf i.e. leaf node array.

2. For leaf node array 1 to n perform the following process

2.1 Perform the bottom up tree traversal for each leaf node to the root and generate all
Combinations of candidate itemset using the function candidate_Generation (Lnode, Item.Lnode) example, from figure3 we start from the first
Pointer P1, candidate 2 itemsets are 12*, 111:1, 12*, 11*:1, 12*, 1**:1.
2.2 Function candidate_Generation (Lnode Arrays 1...N perform as follows. If Lnode is the leaf node and Lnode.item is the item in the leaf node and Cnode is the current node. And cnode.item is the item in current node and SC is the support count in the nodes and SC stores the minimum support count of their parameters.

```
Candidate_Generation (LNode.Arrays 1..N)
{
     Gen_Freq2 (Lnode.Arrays 1...N);
     For Each {cand,SC}€ Cand2
         If SC< Min.Support then Remove (cand, SC) from Cand2
     Gen_Freqk (Lnode.Arrays 1...N)
```

```
  2.3 Gen_Freq2  (Lnode.Arrays 1..N)
   Lnode.Arrays = 1
   While(Lnode.Arrays ≤ N)
    {
     While (Lnode!=Null)
       {
            Cnode = LNode +1
  SC = minimum (Lnode.SC,Cnode.SC)
            Insert to Cand2 ({Lnode.Item,Cnode.item},SC)
Ans_des(Lnode.Item,Cnode.item)
           Lnode = Lnode +1
            }
     Lnode.Arrays = Lnode.Arrays + 1
    }
```

2.4 Function Ans_des(Lnode.Item,Cnode.item) check the Lnode and Cnode for its ancestor. If Lnode has ancestor then generate candidate itemset with ancestor.Lnode.item with Cnode.item and insert it into cand2 ({ancestor.Lnode.item,cnode.item},SC) repeat the same for Cnode. If Cnode has ancestor then generate itemset with Lnode.item and ancestor.Cnode.item and insert them into cand2 ({Lnode.item,ancestor.Cnode.item},SC).

```
Cand2(Cand, SC)
{
          If {cand, SC) € Cand2 then Cand2 = Cand2 U {Cand: SC}
```

```
            Else
              Add SC to the count of the Cand
  }
```

Eliminate the candidate pattern if an item has ancestor and descendent relationship i.e. for Example 1** is the ancestor of 12* so 12*, 1**:1 is eliminated.

```
  Candk(Cand, SC)
{
            If {cand, SC) € Candk then Candk = Candk U
{Cand: SC}
            Else
              Add SC to the count of the Cand
  }
```

2.7 Repeat the algorithm for generating candidate k pattern with support count and stop the process. While generating candidate 3 to k pattern, each node visited will be conserved into buffer for recording the passing path.

For Example the frequent pattern generated through this algorithm is and minimum support count to be satisfied is 3.

2-itemsets	3-itemsets
1,2****	**1**,21**,22***
1,21***	**2**,11*12***
1,22***	**11*,12*,22***
2,11***	**11*,21*,22***
2,12***	**1**,21*,22***
11*,12*	**11*211,22***
11*,21*	**11*,221,12***
11*,22*	**21*,111,22***
12*,22*	**22*,111,211**
21*,22*	
1,211**	
1,221**	
2,111**	
11*,211	
11*,221	
12*,111	
12*,221	
21*,111	
22*,111	
22*,211	
111,211	

The Frequent itemsets derived from this algorithm is similar to ML_T2L1 approach. Since the Frequent 1 pattern is derived through CCB-tree concept which produces the same result as in ML_T2L1.

The advantage of our method over a Level-Crossing [9] and CLFPM [5] is again the minimization of dataset scanning and it generates frequent pattern with single FP-tree and without generating conditional FP-trees. Specifically we can do cross-level construction after building an FP-tree, and then using the bottom up technique to generate the cross-level frequent pattern.

5. PERFORMANCE EVALUATION

We evaluate the performance of our proposed algorithm MCLFPT, Level-crossing [9] and CLFPM [5]. All our experiments were conducted on Intel CPU using Visual Basic Programming Language running in Microsoft windows XP environment. We used Synthetic transactional databases generated by IBM Quest Market-Basket Synthetic Data Generator. We see a comparison between MCLFPT, Level-crossing [9] and CLFPM with short simple transactions. The execution time decreases with MCLFPT, when comparing with Level-crossing and CLFPM

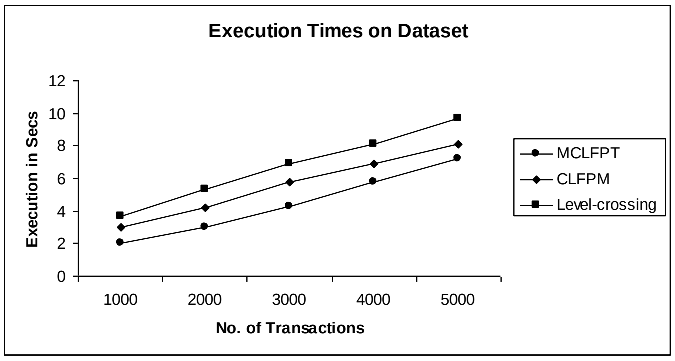

Fig. 5: Performance Evaluation

First, the relative performance of the MCLFPT algorithms under any setting is relatively independent of the number of transactions used in the testing, which indicates that the performance is highly relevant to threshold setting. Second, the MCLFPT algorithm have relatively good 'scale-up' behavior since the increase of the number of the transactions in the database will lead to approximately the linear growth of the processing of large transaction databases. From our experiments, we conclude that MCLFPT is most efficient and stable among all the algorithms based on FP-tree structure. It reduces the I/O Costs.

6. CONCLUSIONS & FUTURE WORK

Transaction databases in many applications contain data that has built-in hierarchy information. In such databases, users may be interested in finding association among items only at the same level and we extended the scope of study of mining level-crossing association rules from large databases. A transaction reduction technique based method is used to reduce the unwanted candidates and transactions and applying the resulted transactions in FP-tree as input to subsequent iterations of the mining process. We adopted a bottom-up approach, with a leaf to root traversal with single FP-tree generation, so as to identify frequent patterns existing between arbitrary classification levels. Our method reduces the I/O costs and search spaces without losing any patterns. Performance Evaluation demonstrates the viability of our new method. In future, an efficient algorithm can be generated to reduce the redundancy in cross-level association rules.

7. REFERENCES

[1] Agrawal R,Imienlinski 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.

[2] Agrawal R, and Srikant R, (1994). Fast algorithms for mining association rules. In Proc. Of the 20th Int. Conf. on very Large Databases. Pages 487-499.

[3] 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.

[4] Han, J., Fu, Y., Mining Multiple-Level Association Rules in Large Databases, in IEEE Transactions on Knowledge and Data Engineering, Vol. 11, No. 5, September/October 1999.

[5] T.Eavis and XI Zheng, Multi-Level Frequent Pattern Mining, in Springer-Verlag Berlin Heidelberg 2009, pp. 369 – 383.

[6] Dr.K.Duraiswamy and B.Jayanthi, a Novel preprocessing Algorithm for Frequent Pattern Mining in Mutidatasets, International Journal of Data Engineering,Vol. 2, No. 3, Aug 2011.

[7] Han, J., Fu, Y., Discovery of Multiple-Level Association Rules from Large Databases, in Proceedings of the 21st Very Large Data Bases Conference, Morgan Kaufmann, P. 420-431, 1995.

[8] Yinbo WAN, Yong LIANG, Liya DING, "Mining Multilevel Association Rules from Primitive Frequent Itemsets", Journal of Macau University of Science and Technology, Vol.3 No.1, 2009

[9] Thakur, R. S., Jain, R. C., Pardasani, K. R., Mining Level-Crossing Association Rules from Large Databases, in the Journal of Computer Science 2(1), P. 76-81, 2006.

[10] R.E.Thevar, R.Krishnamoorthy, A New Approach of Modified Transaction Reduction Algorithm For mining Frequent Itemset, proceedings of IEEE Workshop on Data mining and Artificial Intelligence, 2008.

[11] Rajkumar.N, Karthik.M.R, Sivanada.S.N, "Fast Algorithm for mining multilevel Association Rules,"IEEE Trans. Knowledge and Data Engg., Vol.2 pp. 688-692, 2003.

[12] Pratima Gautham, Pardasani, K. R., "Algorithm for Efficient Multilevel Association Rule Mining", International Journal of Computer Science and Engineering, Vol.2 pp. 1700-1704, 2010