A Fast Algorithm for Mining Multilevel Association Rule Based on Boolean Matrix

[1]Pratima Gautam
Department of computer Applications
MANIT
Bhopal (M.P.)

[2] K. R. Pardasani
Department of Mathematics
MANIT
Bhopal (M.P.)

***Abstract* - In this paper an algorithm is proposed for mining multilevel association rules. A Boolean Matrix based approach has been employed to discover frequent itemsets, the item forming a rule come from different levels. It adopts Boolean relational calculus to discover maximum frequent itemsets at lower level. When using this algorithm first time, it scans the database once and will generate the association rules. Apriori property is used in prune the item sets. It is not necessary to scan the database again; it uses Boolean logical operation to generate the multilevel association rules and also use top-down progressive deepening method.**

Keywords - association rules, data mining, fuzziness, multilevel rules.

I. INTRODUCTION

Data mining, or the efficient discovery of interesting patterns from large collections of data, has been recognized as an important area of database research. The most commonly sought patterns are association rules as introduced in [4]. Association rule mining is an important data mining technique to generate correlation and association rule. The problem of mining association rules could be decomposed into two sub problems, the mining of large itemsets (i.e. frequent itemsets) and the generation of association rules[1] [3]. An association rule is an implication of the form $A \Rightarrow B$, where $A \subseteq I$, $B \subseteq I$, and $A \cap B = \phi$. The rule $A \Rightarrow B$ holds in the transaction set *D,* with support *s*, where *s* is the percentage of transactions in *D* that contain $A \cup B$ (i.e., the *union* of sets *A* and *B*, or say, both *A* and *B*). This is taken to be the probability, $P(A \cup B)$.The rule $A \Rightarrow B$ has confidence *c* in the transaction set *D*, where *c* is the percentage of transactions in *D* containing *A* that also contain *B*. This is taken to be the conditional probability, $P(B|A)$. That is,

$$Support\ (A \Rightarrow B) = P(A \cup B)$$

$$Confidence(A \Rightarrow B) = P(B \mid A)$$

Rules that satisfy both a minimum support threshold (min sup) and a minimum confidence threshold (min conf) are called strong [14]. A set of items is referred to as an itemset. An itemset that contains *k* items is a *k*-itemset. The set of {computer, laser printer} is a 2-itemset. The occurrence frequency of an itemset is the number of transactions that contain the itemset. This is also known, simply, as the frequency, support count, or count of the itemset [13]. Note that the itemset support defined in the equation given below is sometimes referred to as relative support, whereas the occurrence frequency is called the absolute support. If the relative support of an itemset *I* satisfies a prespecified minimum support threshold (i.e., the absolute support of *I* satisfies the corresponding minimum support count threshold), then *I* is a frequent itemset [2]. The set of frequent *k*-itemsets is commonly denoted by L_k.

$$Confidence(A \Rightarrow B) = P(B \mid A) = \frac{Support(A \cup B)}{Support(A)}$$

The problem of mining association rules can be reduced to that of mining frequent itemsets [10]. Association rule mining can be viewed as a two-step process:

1. Find all frequent itemsets: By definition, each of these itemsets will occur at least as frequently as a predetermined minimum support count, *min sup.*

2. Generate strong association rules from the frequent itemsets: By definition, these rules must satisfy minimum support and minimum confidence.

It is difficult to find strong and interesting associations among data items at the primitive levels of abstraction due to the paucity of data.

However, many strong associations discovered at rather high concept levels are common sense knowledge. Therefore, a mining system with the capabilities to mine association rules at multiple levels of abstraction and traverse easily among different abstraction spaces is more desirable like Han et al. in [15] and Rajkumar et al. in [8] indicate.

We are using multilevel association rule and Boolean association rule in our algorithm called MLBM. Boolean association rule mining is used more widely than other kinds of association rule mining [9]. This algorithm transforms a transaction database into a Boolean matrix stored in bits. Meanwhile it uses the Boolean vector "relational calculus" method to discover frequent itemsets. We use the fast and simple "and calculus'' in the Boolean matrix to replace the calculations and complicated transactions that deal with large numbers of itemsets [5]. This algorithm is more effective than the Apriori-like algorithms. This algorithm is also used progressive deepening method. The method first finds frequent data items at the top most level and then progressively deepens the mining process into their frequent descendants at lower concept levels [7]. This method is using concept of reduced support and refine the transaction able at each level.

II. APORIORI ALGORITHM

The key of mining association rules is to set an appropriate support and confidence values to find frequent itemset. The well-known algorithm, Apriori, exploits the following property: If an itemset is frequent, so are all its subsets [1]. Apriori employs an iterative approach known as level wise search, where k -itemsets are used to explore k 1-itemsets. First, the set of frequent 1-itemsets is found. This is denoted as L_1. L_1 is used to find L_2, the frequent 2-itemsets, which is used to find L_3, and so on, until no more frequent k -itemsets can be found. The finding of each L_k requires one full scan of the database. Throughout the level-wise generation of frequent itemsets, an important anti-monotone heuristic is being used to reduce the search space [16].

III. MULTI LEVEL ASSOCIATION RULE MINING

We can mine multilevel association rules efficiently using concept hierarchies, which defines a sequence of mappings from a set of low-level concepts to higher-level, more general concepts [6] [17]. Data can be generalized by replacing low-level concepts within the data by their higher-level concepts or ancestors from a concept hierarchy. In a concept hierarchy, which is represented as a tree with the root as D i.e., Task-relevant data. The popular area of application for multi level association is market basket analysis [8] [11], which studies the buying habits of customers by searching for sets of items that are frequently, purchased together which was presented in terms of concept hierarchy shown below. Each node indicates an item or item set that has been examined. There are various approaches for finding frequent item sets at any level of abstraction. Some of the methods which are in use are 'using uniform minimum support for all levels', using reduced minimum support at low levels, level-by-level independent.

Multi-level databases use hierarchy-information encoded transaction table instead of the original transaction table [7]. This is useful when we are interested in only a portion of the transaction database such as food, instead of all the items. This way we can first collect the relevant set of data and then work repeatedly on the task-relevant set. Thus in the transaction table each item is encoded as a sequence of digits.

Example: encoded as a sequence of digits in the transaction table T [1]. For example, the item `2 percent foremost milk' is encoded as `112' in which the first digit, `1', represents `milk' at level-1, the second, `1', for `2 percent (milk)' at level-2, and the third, `2', for the brand `Foremost' at level-3. Similar to [15], repeated items (i.e., items with the same encoding) at any level will be treated as one item in one transaction.

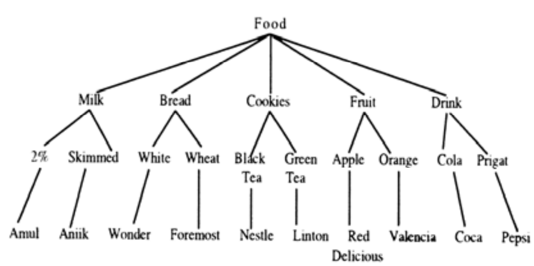

Fig1. The taxonomy for the relevant data items

IV. A MULTILEVEL ALGORITHM BASED ON BOOLEAN MATRIX (MLBM)

We propose a new multilevel association algorithm. The section is organized as follows: the correlative definition and proposition, an introduction to the MLBM algorithm details, and a description of a sample execution of the MLBM algorithm.

A. Definition and proposition

Association Rules

Definition 1: Let I = {i_1 , i_2, i_3,……. i_n} be a set of items. D is a database of transactions. Each transaction T is a set of items and has an identifier called TID. Each T$\subseteq$I. [9]

Definition 2: Association rule is the implication of the form $A \Rightarrow B$, where A and B are itemsets which satisfies A$\subseteq$I, B$\subseteq$I and A$\cap$ B =ϕ .

Definition 3: The strength of an association rule can be measured in terms of its *Support* and *Confidence.* Rule $A \Rightarrow B$" is true in *D* with a support (denoted by "*sup*") and a confidence (denoted by "*conf*"), where *A* and *B* are set of items. Support "*sup*" is a percentage of transactions including both *A* and *B* ($A \cup B$) in transaction sets *D*. Confidence "*conf*" is a percentage of transactions including both *A* and B ($A\cup B$) in transactions that contain *A*.[12]

Sup=P ($A\cup B$)/|D|, $conf$ =P ($B|A$) $=P$ ($A\cup B$)/P (A)

Definition 4: Boolean Matrix: is a matrix with element '0' or '1'.

Definition 5: The Boolean AND operation is defined as follows:

0.0=0 0.1=0 1.0=0 1.1=1

Where logical implication is denoted by '**.**' or AND. If we write C=A.B, then C can be determined by listing all possible combinations of A and B. Truth table for logical AND will be:

TABLE I . AND OPERATOR

A	B	C=A.B
0	0	0
0	1	0
1	0	0
1	1	1

Definition 6: The Boolean 'AND calculus' is carried out to an arbitrary k columns vector of the Boolean matrix; the sum of '1' of the operation result is called k- support of the k columns vector.

Proposition 1: If the sum of '1' in a column vector A_i is less than min_sup_num, it is not necessary that A_i will attend calculus of the next level supports.

Rationale: According to the principle of the Boolean AND calculus, the result is '1' when the value of all vector elements (in a record) is '1' [5].

Proposition 2: Itemset A is a k-itemsets (each item belongs to different level); |LK-1(j)| presents the number of values in a level 'j' in all frequent (k-1)-itemsets of the frequent set LK-1. There is an item j in X. If |LK-1(j)| is smaller than k-1, itemset X is not a frequent itemset[5] .

b. Algorithm Details (MLBM)

The algorithm consists of following steps:

Step-1:
Encode taxonomy using a sequence of numbers and the symbol ''*'', with the ***l***th number representing the branch number of a certain item at levels.

Step-2:
Set H = 1, where H is used to store the level number being processed whereas H $\in${1, 2, 3} (as we consider up to 3-levels of hierarchies).

Step-3:
Transforming the transaction database into the Boolean matrix.

Step-4:
Set user defines minimum support on current level.

Step-5
Generating the set of frequent 1-itemset L_1at level 1.

Step-5:
Pruning the Boolean matrix

Step-6:
Perform AND operations to generate 2-itemsets and 3- itemset at level 1.

Step-7:

Generate H +1; (Increment H value by 1; i.e., H = 2) itemset from L_k and go to step-4 (for repeating the whole processing for next level).

c. Transforming the transaction database into the Boolean matrix

The mined transaction database is D, with D having m transactions and n items. Let T= {T_1, T2,…, Tm} be the set of transactions and I= {I_1, I_2,…,I_n) be the set of items. We set up a Boolean matrix A_{m*n}, which has m rows and n columns. Scanning the transaction database D, if item I_j is in transaction T_i , where 1≤j≤n the element value of $A_i$is '1,' otherwise the value of I_j is '0.'

d. Generating the set of frequent 1-itemset L_1

The Boolean matrix A_{m*n} is scanned and support numbers of all items are computed. The support number I_j.supth of item I_j is the number of '1s' in the jth column of the Boolean matrix A_{m*n}. If I_j.supth is smaller than the user define minimum support number *minsupth*, itemset {I_j} is not a frequent 1-itemset and the jth column of the Boolean matrix A_{m*n} will be deleted from A_{m*n}.

e. Pruning the Boolean matrix

Pruning the Boolean matrix means deleting some columns from it. This is described in detail as: Let

I′ be the set of all items in the frequent set LK-1, where k>2. Compute all |LK-1(j)| where j∈I′, and delete the column of correspondence item j if |LK-1(j)| is smaller than min_sup_num.

f. Generating the set of frequent k-itemsets L_k

Frequent k-itemsets are discovered by AND relational calculus, which is carried out for the k-vectors combination. If the Boolean matrix Ap*q has q columns where 2<q≤n and *min_sup_num* is h≤p≤m, $(C_q)^k$, combinations of k-vectors will be produced. The AND relational calculus is for each combination of k-vectors. If the sum of element's values in the 'AND' calculation result is not smaller than the minimum support number *min_sup_num*, the k-itemsets corresponding to this combination of k-vectors are the frequent k-itemsets and are added to the set of frequent k-itemsets L_k.

5. An Illustrative Example:

An illustrative example is given to understand well the concept of the proposed algorithm and how the process of the generating multilevel

association rule mining is performed step by step. The process is started from a given transactional database as shown in Table 1[a].

Table 1[a]

Trans_ID	***List of items***
T1	**111, 212, 311**
T2	**111, 222, 311, 411, 511**
T3	**111, 222, 411**
T4	**121, 321, 422,521**
T5	**111, 222, 311, 411**
T6	**222, 311, 422**

Table.1 [a] Transaction data of the transaction database D.

Table1 [b]
Codes of item name

Code	***Description***
1**	Milk
2**	Bread
3**	Cookies
4**	Fruit
5**	Drink
11*	2%
12*	Skimmed
21*	White
22*	Wheat
31*	Black Tea
32*	Green Tea
41*	Apple
42*	Orange
51*	Cola
52 *	Drink Prigat
111	Milk 2% Amul
121	Milk Skimmed Anik
211	Bread White Wonder
222	Bread wheat Foremost
311	Cookies black Tea Nestle
321	Cookies Green Tea Linton
411	Fruit Apple red Delicious
422	Fruit Orange Valencla
511	Drink cola Coca
522	Drink Prigat pepsi

The transaction database D is transformed into the Boolean matrix A6*5:

Fig. 2 The Boolean matrix A6*5

We execute the MLBM algorithm at level-1. Therefore minimum support number = 3.0

$$
\begin{array}{c c}
 & \begin{array}{ccccc} 1^{**} & 2^{**} & 3^{**} & 4^{**} & 5^{**} \end{array} \\
\begin{array}{c} T1 \\ T2 \\ T3 \\ T4 \\ T5 \\ T6 \end{array} &
\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix}
\end{array}
$$

We compute the sum of the element values of each column in the Boolean matrix A6*5 and the set of frequent 1-itemset is:

{{1**}, {2**}, {3**}, {4**}}

In pruning the Boolean matrix A6*5 .The fifth column of the Boolean matrix A6*5 is deleted because the support number of item 5** is smaller than the minimum support number. Finally, the Boolean matrix A6*4 is generated.

$$
\begin{array}{c c}
 & \begin{array}{cccc} 1^{**} & 2^{**} & 3^{**} & 4^{**} \end{array} \\
\begin{array}{c} T1 \\ T2 \\ T3 \\ T4 \\ T5 \\ T6 \end{array} &
\begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}
\end{array}
$$

We perform the 'AND' operation to generate 2-itemset at level-1.And now matrix is A_{6*6}.

The possible 2-itemsets are: **(1** ∧ 2**), (1** ∧ 3**), (1** ∧ 4**), (2** ∧ 3**), (2** ∧ 4**), (3** ∧ 4**)**

$$
\begin{array}{c} \\ T1 \\ T2 \\ T3 \\ T4 \\ T5 \\ T6 \end{array}
\begin{array}{c}
\begin{array}{cccccc} (1^{**}\wedge 2^{**}) & (1^{**}\wedge 3^{**}) & (1^{**}\wedge 4^{**}) & (2^{**}\wedge 3^{**}) & (2^{**}\wedge 4^{**}) & (3^{**}\wedge 4^{**}) \end{array} \\
\begin{bmatrix}
1 & 1 & 0 & 1 & 0 & 0 \\
1 & 1 & 1 & 1 & & 1 \\
1 & 0 & 1 & 0 & 1 & 0 \\
0 & 1 & 1 & 0 & 0 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 1 & 1 & 1
\end{bmatrix}
\end{array}
$$

We compute the sum of the element values of each column in the Boolean matrix A_{6*6} and all 2-itemset considered for further process because their support numbers are greater than the minimum support number. Again we perform 'AND'operation to generate 3-itemset and finally matrix is A_{6*4}.

The possible 3-itemsets are: **(1** ∧ 2** ∧ 3**) (1** ∧ 2** ∧ 4**)(1** ∧ 3** ∧ 4**) (2** ∧ 3** ∧ 4**)**

$$
\begin{array}{c} \\ T1 \\ T2 \\ T3 \\ T4 \\ T5 \\ T6 \end{array}
\begin{array}{c}
\begin{array}{cccc} (1^{**}\wedge 2^{**}\wedge 3^{**}) & (1^{**}\wedge 2^{**}\wedge 4^{**}) & (1^{**}\wedge 3^{**}\wedge 4^{**}) & (2^{**}\wedge 3^{**}\wedge 4^{**}) \end{array} \\
\begin{bmatrix}
1 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
1 & 1 & 1 & 1 \\
0 & 0 & 0 & 1
\end{bmatrix}
\end{array}
$$

We compute the sum of the element values of each column in the Boolean matrix A_{6*4} and all 3-itemset considered for further process because their support numbers are greater than the minimum support number and we go to next level.

Level-2

Minimum_support = 2.0

1-itemset

$$
\begin{array}{c} \\ \\ T1 \\ T2 \\ T3 \\ T4 \\ T5 \\ T6 \end{array}
\begin{array}{c}
\begin{array}{cccc} \underbrace{1\,*\,*} & \underbrace{2\,*\,*} & \underbrace{3\,*\,*} & \underbrace{4\,*\,*} \end{array} \\
\begin{array}{cccccccc} 11^* & 12^* & 21^* & 22^* & 31^* & 32^* & 41^* & 42^* \end{array} \\
\begin{bmatrix}
1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\
1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\
1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 1 & 0 & 0 & 1
\end{bmatrix}
\end{array}
$$

We compute the sum of the element values of each column in the Boolean matrix A_{6*8} and 12*, 21*, 32* column are deleted because their support numbers are smaller than the minimum support number. Again perform 'AND' operation to generate 2-itemset at level 2.Finally, the Boolean matrix $A_{6*9}$is generated.

The possible 2-itemsets are: **(11* ∧ 22*) (11* ∧ 31*) (11* ∧ 41*) (11* ∧ 42*) (22* ∧ 31*) (22* ∧ 41*) (22* ∧ 42*) (31* ∧ 41*) (31* ∧ 42*)**

$$
\begin{array}{c} \\ T1 \\ T2 \\ T3 \\ T4 \\ T5 \\ T6 \end{array}
\begin{array}{c}
\begin{array}{ccccccccc} (11^*\wedge 22^*) & (11^*\wedge 31^*) & (11^*\wedge 41^*) & (11^*\wedge 42^*) & (22^*\wedge 31^*) & (22^*\wedge 41^*) & (22^*\wedge 42^*) & (31^*\wedge 41^*) & (31^*\wedge 42^*) \end{array} \\
\begin{bmatrix}
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\
1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1
\end{bmatrix}
\end{array}
$$

We compute the sum of the element values of each column in the Boolean matrix A6*9. (11* ∧ 42*)(22* ∧ 42*)(31* ∧ 42*) column are deleted because their support numbers are smaller than the minimum support number. Again we perform 'AND' operation to generate 3-itemset and finally matrix is A_{6*3} generated.

The possible 3-itemsets are: **(11* ∧ 22* ∧ 31*) (11* ∧ 22* ∧ 41*) (22* ∧ 31* ∧ 41*)**

(11* ∧ 22* ∧ 31*) (11* ∧ 22* ∧ 41*) (22* ∧ 31* ∧ 41)

$$\begin{array}{c} T1 \\ T2 \\ T3 \\ T4 \\ T5 \\ T6 \end{array} \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

We compute the sum of the element values of each column in the Boolean matrix A_{6*3} and all 3itemset considered for further process because their support numbers are greater than the minimum support number and we go to next level. Finally matrix is A_{6*4}.

Level-3
Minimum_support = 2.0
1-itemset

$$\begin{array}{c} \\ T1 \\ T2 \\ T3 \\ T4 \\ T5 \\ T6 \end{array} \begin{array}{c} \begin{array}{cccc} 111 & 222 & 311 & 411 \end{array} \\ \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

We compute the sum of the element values of each column in the Boolean matrix A_{6*3} and all 3-itemset considered for further process because their support numbers are greater than the minimum support number. Now we perform 'AND' operation on 1-itemset at level-3 and generate 2-itemset, finally matrix is A_{6*6}.

The possible 2-itemsets are: **(111 ∧ 222) (111 ∧ 311) (111 ∧ 411) (222 ∧ 311) (222 ∧ 411) (311 ∧ 411)**

$$\begin{array}{c} \\ T1 \\ T2 \\ T3 \\ T4 \\ T5 \\ T6 \end{array} \begin{array}{c} \begin{array}{cccccc} (111 \wedge 222) & (111 \wedge 311) & (111 \wedge 411) & (222 \wedge 311) & (222 \wedge 411) & (311 \wedge 411) \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \end{array}$$

We compute the sum of the element values of each column in the Boolean matrix A_{6*6} and all 2-itemset considered for further process because their support numbers are greater than the minimum support number and Now we perform 'AND' operation on 2-itemset to generate 3-itemset at level-3. Finally matrix is A_{6*3} generated.

$$\begin{array}{c} \\ T1 \\ T2 \\ T3 \\ T4 \\ T5 \\ T6 \end{array} \begin{array}{c} \begin{array}{ccc} (111 \wedge 222 \wedge 311) & (111 \wedge 222 \wedge 411) & (222 \wedge 311 \wedge 411) \end{array} \\ \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix} \end{array}$$

T2

According to step-3, the MLBM algorithm is terminated because there are maximum frequent itemset find at lower level(level3).

CONCLUSION

In this paper, a multilevel association rule mining algorithm based on the Boolean matrix (MLBM) is proposed. The main features of this algorithm are that it only scans the transaction database once, it does not produce itemsets, and it adopts the Boolean vector "relational calculus" to discover frequent itemset. In addition, it stores all transaction data in bits, so it needs less memory space and can be applied to mining large transaction databases.

Reference:

[1] R. Agrawal, T. Imielinski, and A. Swami, "Mining association rules between sets of items in large databases," Proceedings of the ACM SIGMOD Conference on Management of data, pp. 207-216, 1993.
[2]R. Agrawal and R. Srikant," Fast algorithms for mining association rules," In proceeding of the VLDB Conference,1994.
[3] H. Mannila, H. Toivonen, and A, Verkamo. "Efficient algorithm for discovering association rules," AAA1 Workshop on Knowledge Discovery in Databases.
[4]Jiawei Han, Micheline Kamber, "Data Mining Concepts and Techniques," Higher Education Press 2001.
[5] Hunbing Liu and Baishen wang, "An association Rule Mining Algorithm Based On a Boolean Matrix," Data Science Journal, Vol-6, Supplement 9, S559-563, September 2007.
[6] R.S Thakur, R.C. Jain, K.R.Pardasani, "Fast Algorithm for Mining Multilevel Association Rule Mining," Journal of Computer Science, Vol-1, pp. 76-81, 2007 .
[7] Ha and Y. Fu, "Mining Multiple-Level Association Rules in Large Databases," IEEE TKDE. Vol-1, pp. 798-805, 1999 .

[8] N.Rajkumar, M.R.Karthik, S.N.Sivana and S.N.Sivanandamndam, "Fast Algorithm for Mining Multilevel Association Rules," IEEE, Vol-2, pp. 688- 692, 2003.

[9] Maurice Houtsma and A , Swam I, "Set-oriented mining of association rules," Research Report RJ 9567, IBM Almaden Research Center[C], San Jose, Califomia:[s.n.] 1993

[10] A.K.H. Tung, H. Lu, J. Han and L, Feng, "Efficient mining of intertransaction association rules," *IEEE Trans.on Knowledge and Data Engineering*, vol-15, no. 1, pp.43-56, Jan./Feb. 2003.

[11] S. Brin, R. Motwani, J. D. Ullman and S. Tsur, "Dynamic itemset counting and implication rules for market basket data," *ACM SIGMOD International Conference on Management of Data*, pp. 255–264, May 1997.

[12] Anjna Pandey and K. R. Pardasani,"Rough Set Model for Discovering Hybrid Dimensional Association Rules," International Journal of Computer Science and Network Security, Vol -9, no.6, pp.159-164,2009.

[13] Ravindra Patel, D. K. Swami, K. R. Pardasani, "Lattice Based Algorithm for Incremental Mining of Association Rules," International Journal of Theoretical and Applied Computer Sciences, Vol- 1, pp. 119–128, 2006.

[14] Jun Gao ,"Realization of a New Association Rule Mining Algorithm," IEEE DOI, 2007.

[15] Scott Fortin and Ling Liu,"An object-oriented approach to multi-level association rule mining," Proceedings of the fifth international conference on Information and knowledge management, pp.65 72 1996.

[16] Neelu Khare , Neeru Adlakha and K. R. Pardasani "Karnaugh Map Model for Mining Association Rules in Large Databases," *(IJCNS) International Journal of Computer and Network Security,Vol- 1, No. 1, October 2009*

[17] Pratima Gautam Neelu Khare and K. R. Pardasani, "A model for mining multilevel fuzzy association rule in database," Journal of computing, vol- 2, issue- 1, pp. 58-68 January 2010