ISSN No. 0976-5697

Volume 3, No. 3, May-June 2012

International Journal of Advanced Research in Computer Science

REVIEW ARTICLE

Available Online at www.ijarcs.info

A Review of Various Methods for Association Rule Mining

Harpreet Singh*
Department of Computer Science and Engineering
National Institute of Technology
Jalandhar, India
harpreet99.nitj@gmail.com

Renu Dhir
Department of Computer Science and Engineering
National Institute of Technology
Jalandhar, India
dhirr@nitj.ac.in

Munish Saini
Department of Computer Science and Engineering
National institute of Technology
Jalandhar, India
munish_1_saini@yahoo.co.in

Abstract**—**Association rule mining is the process of finding interesting relationships between various data elements. As the size of database is growing so rapidly, efficient methods are required for finding association rules. This paper presents a review of various known and recently developed methods for finding association rules. Followed by the comparison of different methods and specifying which method is more efficient from other.

Keywords**—** Apriori algorithm; Association rule; Frequent itemsets; QFP; MFP; APFT;FP-Split, Pattern Matrix.

I. INTRODUCTION

Data mining is one of the key research areas which have attracted the attention of various researchers and practioners. Due to the wide availability of data, the need for turning such data into useful information and knowledge is generated. The useful information obtained can be applied to various applications ranging from market basket analysis, fraud detection, customer retention, early warning of equipment failure etc. Hence, Data mining [1] refers to the process of extracting knowledge or useful data patterns from large amount of data by applying various intelligent methods.

Association rule mining is one of the key research areas of the data mining. In this paper Section II will give the description of Association rule mining. Section III and Section IV will present the review of various known methods and recently developed methods for association rule mining respectively. Section V will give the comparison of various methods.

II. DESCRIPTION OF THE ASSOCIATION RULE MINING

Association rule mining [1] is the process of finding useful relationship between different data items of the large database and then representing this relationship in the form of rules called as Association rules. Association rules [1] can be written as

$X \rightarrow Y$ [Support= s%, Confidence=c%]

Support s, is the probability that a transaction contains (X, Y).

$$\text{Support } (X \rightarrow Y) = P\ (XUY),$$

Confidence c, is the conditional probability that a transaction contain X also contain Y.

$$\text{Confidence}(X \rightarrow Y) = P(Y/X) = P\ (XUY)/\text{support\_count}(X)$$

Support and Confidence are the two measures of rule interestingness. Support and confidence represent the usefulness and certainty of the discovered rules. Minimum support and Minimum confidence are needed to eliminate the unimportant rules. So, the association rule holds if its support and confidence value is greater than minimum support and minimum confidence values and such rules are called as the interesting rules.

III. REVIEW OF VARIOUS KNOWN METHODS

A. *Apirori Algorithm:*

Apriori Algorithm [1, 2] is one of the classical algorithm proposed by R. Srikant and R. Agrawal in 1994 for finding frequent patterns for Boolean association rules. 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-itemset L_1 is found. Next, L1 is used find frequent 2-itemset L_2. Then L_2 is used to find frequent 3-itemset L_3. The method iterates like this till no more frequent k-itemsets are found.

Apriori Algorithm finds frequent itemsets from candidate itemsets. It is executed in two steps; firstly it retrieves all the frequent itemsets from the database by considering those itemsets whose support is not smaller than the minimum support (min_sup). Secondly, it generates the association rules satisfying the minimum confidence (min_conf) from the frequent itemset generated in first step. The first step consists of join and pruning action. While joining the candidate set C_k is produced by joining L_{k-1} with itself and pruning the candidate sets by applying the Apriori property i.e. All the non-empty subset of frequent itemset must also be frequent.

The pseudo code for generation of frequent itemsets is given below.

C_k: Candidate itemset of size k

L_k: Frequent itemset of size k

```
{
    L1= frequent 1-itemset
    For (k=1; k! =NULL; k++)
      {
        Ck+1=Join Lk with Lk to generate Ck+1;
         Lk+1= Candidate in Ck+1 with support greater than
                    or equal to min support;
      }
     End;
Return Lk;
}
```

B. fp- Growth Algorithm:

FP-Growth algorithm [1,4,5] proposed by Jiawei Han finds the association rules more efficiently than Apriori algorithm without the generation of candidate itemsets. Apriori algorithm requires n+1 scans, where n is the length of the longest pattern. FP-Growth algorithm requires only two scans of the database to find frequent patterns. FP-Growth algorithm adopts divide and conquer strategy. First, it construct a FP-tree [5] using the data in transactional database and then mines all the frequent patterns from FP-tree. After mining of frequent patterns the association rules can be generated easily.

The pseudo code for FP-Growth algorithm is as follows [1, 3]

a. If Tree contains a single path P THEN
b. for all combination (denoted as β) to the nodes in path P Do
c. Generate pattern β U α with support = minimum support of nodes in β;
d. else for each a_i in the heads of tree Do
e. Generate pattern β = a_i U α with support= a_i . support
f. Construct conditional pattern base and generate FP-tree $Tree_\beta$
g. If $Tree_\beta$= Φ THEN
h. Call FP-Growth ($Tree_\beta$ β)
i. End

C. Partitioning Method:

Partitioning Method [1] provide the improvement over classical Apriori algorithm. It works in two steps. In first step, it divides the transactions of the database D into n non-overlapping partitions and then finds the support count of each partition. And in second step, global frequent itemset among the candidates is found. The Partitioning method requires only two database scan as compare to n+1 scans required by the Apriori algorithm. The process of Partitioning method is shown below in Figure1.

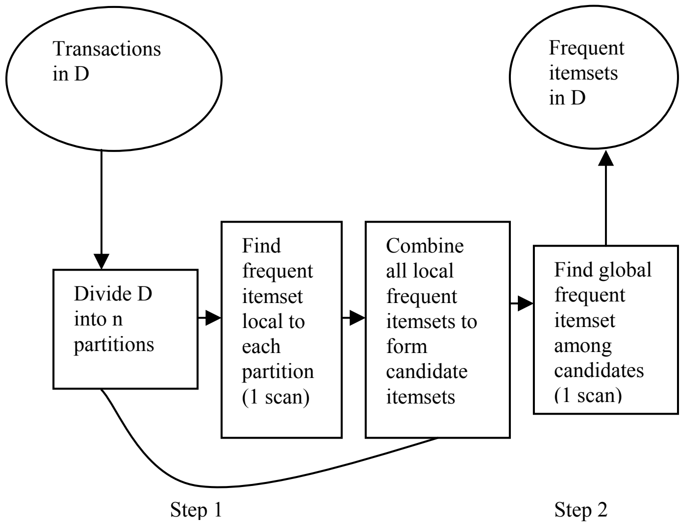

Figure 1. Mining of frequent itemsets by Partitioning the data

D. Transaction Reduction Method:

Transaction Reduction Method [1] employs a property that a transaction that doesn't contains any frequent k-itemsets cannot contain any frequent (k+1)-itemset. Therefore, such transaction can be removed from the database for further consideration.

The pseudo code for Transaction Reduction Method is as follows

a. Scan the database D to find L_k frequent itemset
b. For L_{k+1} Do
c. If transaction does not contain any frequent k-itemset Then
d. Delete the transaction t_k from database D for further consideration
e. Else
f. Consider the transaction
g. End;

The experimental results obtained from various researchers have shown that the transaction reduction method requires less database scans and comparisons as compare to Apriori algorithm.

E. Hashing Method:

Hashing is the method to improve the efficiency of Apriori algorithm. In Hashing Technique [1] the frequent itemsets are found by mapping the frequent items into hash buckets of hashing table. Hashing technique can reduce the size of candidate k-itemset C_k. For example, when scanning each transaction in the database to generate the frequent 1-itemset C_1, we can generate all the 2-itemset for each transaction, map them into different buckets of a hash table structure and increase the corresponding bucket count. An itemset whose bucket count in hash table is below the minimum support count value cannot be frequent and can be removed from the candidate set.

IV. REVIEW OF RECENTLY DEVELOPED METHODS AND ALGORITHMS

A. A modified Apriori Algorithm with its Applications in Instituting Cross-Selling Strategies of the Retail Industry:

Changsheng Zhang and Jing Ruan [6] presented an improved Apriori algorithm with its application in cross selling strategies of retail industry. They proposed an optimized method for Apriori algorithm. The proposed method introduced more efficient way to achieve the pruning operation. The algorithm needs to scan L_{k-1} one time to complete the deletion and remaining of each element X in C_k. In Apriori algorithm L_k is generated from candidate itemset C_k by scanning the database and by calculating each candidate support count. Most of the improved algorithm will first generate (k-1)-item subset of each element X in C_k and compare with $L_{k-1.}$ If a (k-1)-itemset is not the element of L_{k-1}then it is not frequent itemset. According to the Apriori property, X is not frequent either. So X could be deleted from $C_{k.}$ This algorithm requires searching L_{k-1} for each element X in C_k. So the main focus of this new method is to reduce the number of candidate itemsets generated and to reduce the I/O spending.

For reduction of candidate itemsets the new algorithm uses the property:

a. T_k is a k-dimensional itemset. If (k-1) subset of k-itemset is not frequent then k-itemset is also not frequent.

For the reduction of I/O spending the proposed algorithm uses the property:

b. If T is a transaction record in database D. if the number of valid data in T is less than k. then, we will not find any element X of frequent itemset L_k in T.

Hence, by using both of the above two properties the modified algorithm can mine the association rules more efficiently and effectively from large database and improves the performance of Apriori algorithm.

B. Reduced Apriori Algorithm with Tag (RAAT):

Wanjun yu, Xiao chun wang and et.al [7] proprosed a Novel algorithm called as Reduced Apriori Algorithm with Tag (RAAT). The prorposed algorithm reduces the number of candidate itemset produced in pruning operation of C_2 and thus improves the efficiency and saves time. The algorithm RAAT optimize subset operation by using transaction tag to speed up support calculation. The experimential results of [7] shows that the RAAT algorithm gives better result in terms of candidate generation and counting the support using database as compare classical Apirori algorithm.

C. An improved apriori algorithm for early warning of equipment failure:

Liu Jing and et.al [8] presented a new method for improving the performance of Apriori algorithm. In this algorithm the items which cananot become the frequent items are deleted in advance. After the first traversal, the support count is counted by set F_k , the committing filter obtained by use of L_{k-1}. The items whose suuport value is less than the minimum support value are deleted from the database. Then the transaction reduction method, the transaction which has number of items less than k-1 are deleted so as to reduce F_k is applied. The experiment performed by [8] shows that the improved algorithm gives better results and overcome the two limitations of Apriori algorithm i.e the number of candidate itemset generated and number of times database is scaned.

D. Barrel Structure Method:

Dongme Sun and et.al [9] present a new algorithm to improve the effectiveness of Apriori algorithm. In this algorithm the researcher used the combination of reverse and forward scan of database to find the maximal frequent itemset [1]. In this algorithm they used the concept of dynamic itemset counting and use the barrel structure [9] to store all the frequent itemsets. In this first, L_k maximal frequent itemset is found along with its support. After this next frequent itemsets are mined i.e. L_{k-1} and their respective support value is counted by using database D. Similarly all the frequent itemset are mined in this way. Then all these frequent itemsets are placed in bit-matrix [10] to count their respective support values. The results of [9] shows that the improved barrel structure method requires very less time for scanning the database as compare to Apriori algorithm and saves the space as it does not produce large number of candidate itemsets.

E. An Implementation of Improved Apriori algorithm [13]:

In this algorithm the concept of support transaction [11] and descending power subset [11] are used. The support of candidate (k+1)-itemset is found by using support transaction of frequent k-itemset. Then, items whose support value is not less than minimum support threshold are considered as frequent itemsets. It avoid scanning resource database repeatedly, i.e. it reduces the number of times the database is scanned as the transactions of all descending power subset supported some itemset support same itemset. The experimental result performed by Gangyang and et.al shows that it requires less space of memory and reduces the frequency of I/O as compare to previous Apriori algorithm.

The Pseudo code for improved algorithm [11] is as follows:

```
L1 = ( big itemset 1);
For ( k=2; Lk-1 != a; k++ ) Do begin
 Ck = apriori.gen (Lk-1 );
   For all itemset C € Ck Do begin;
      C.Support transaction set = {all items};
      For drop exponent subset of all C S Do;
          C.Support transaction set = C.Support transaction
                    set ∩ S.support transaction set;
       C.count = cnt (C.support transaction set);
   End;
  Lk= {C  € Ck | C.count >= minsup };
End;
Return Uk Lk;
```

F. Qfp Algorithm:

Li Juan and Ming De-ting [12] proposed a new method called QFP algorithm. It is an improvement over FP-Growth algorithm. QFP algorithm requires only one database scan to convert the transaction database into QFP tree after data preprocessing. Then directly generates the association rules from the QFP-tree [12] without looking the transaction database.

This algorithm works in two steps:

a. Construction of QFP-Tree
b. Mine the QFP-Tree to obtain the frequent patterns.

The experimental result of QFP algorithm [12] has shown that time efficiency of the QFP algorithm is higher that that of FP-Growth algorithm. The QFP algorithm can be applied to any situation which is suitable for FP-Growth or Apriori algorithm as the input to QFP is same as that of FP-Growth or Apriori algorithm.

G. APFT algorithm [13]:

Qihua Lan, Defu Zhang, Bo Wu given a new method called APFT which combines the Apriori algorithm and FP-Growth algorithm. APFT algorithm still apply divide and conquer strategy of FP-Growth algorithm for mining process. In APFT, the compressed FP-tree is patitioned off a set of conditional subtree, each of the conditional subtree associated with frequent item.

APFT algorithm works on two steps:

a. To construct FP-tree as FP-Growth algorithm do.
b. To use Apriori algorithm to mine the FP-tree.

In second step an additional table called Node table is required which has two fields.

Item-name: specify the name of the node that apperas in the FPT_i.

Item-support : specify the number of node appear with I_i.

The Results of [13] shows that the APFT algorithm work much faster than Apriori algorithm and work still faster than FP-Growth algorithm when minimum support value is small.

H. FP-Split method [14]:

FP-Split method is proposed by Chin-Feng LEE and Tsung-Hsien Shen in 2005. FP-Split algorithm is proposed for improving the performance of FP-Growth algorithm. Many researchers has tried to improve the performance of FP-Growth algorithm but they ignored the fact that time taken to construct the FP-tree is very large. So, Chin-Feng and et.al consider this point and gives a new method called as FP-Split method.

FP-Split method works in three steps:

a. Construction of equivalence class [14] by scanning database.
b. Count the support of each item and filter out non-frequent itemsets.
c. Constructing the FP-Split tree [14] using equivalence class of frequent itemsets.

The particular node structure of FP-Split tree is shown below in Figure2. In FP-Split algorithm the database is scanned only once at the time of creating equivalence class. The time taken to construct FP-Split tree is much less than the time taken for construction of FP-tree. After the Construction of FP-Split tree the FP-Growth algorithm is applied to find the frequent patterns.

<table>
<tr><td>Content</td><td>Count</td><td rowspan="2">Link_sibling</td></tr>
<tr><td colspan="2">List</td></tr>
<tr><td colspan="3">Link_child</td></tr>
</table>

Figure: 2 Node structure of FP-Split tree

I. Mfp Algorithm [15]:

MFP algorithm is an improvement over FP-Growth algorithm. FP-Growth algorithm requires two database scans one for construction of table L and second for construction of FP-tree. But in case of MFP algorithm only one database scan is required.

MFP Algorithm consist of two main steps:

a. Construction of MFP-tree [15]
b. Mining of frequent patterns from MFP-tree.

In MFP-tree each node expect the root node and leaf node has two enerties.

a. Support count value of node
b. Pointer to the next node in MFP-tree.

The results of [15] has shown that MFP algorithm requires less time and can find the frequent patterns by scanning the database only once. This algorithm can be appiled to any situation where FP-Growth or Apriori algorithm is suitable.

J. P_ Matrix Alorithm [16]:

Sixue Bai, Xinxi Dai proposed an efficient and fast algorithm based on Pattern Matrix [16].

The proposed algorithm works in two steps

a. Scan the database once to obtain the binary pattern matrix and transform its ranks. The pattern matrix P is written as shown below in Figure 3.
b. Perform the operation of AND with each row of pattern matrix to generate frequent itemsets.

The P_Matrix algorithm covers both the problems of Apriori algorithm. It scans the database only once to generate pattern matrix and then directly finds the frequent itemsets from the pattern matrix without generation of candidate itemsets. The result of [16] has shown that P_Matrix algorithm greatly reduces the temporal complexity and spatial complexity of the algorithm and improves the efficiency of Apriori algorithm.

$$
\begin{array}{cc|cccccc|}
 & & I1 & I2 & Ij \ldots & Ik \ldots & Im & \\
 & T1 & y_{11} & y_{12} & y_{1j} & y_{1k} & y_{1m} & \\
 & T2 & y_{21} & y_{22} & y_{2j} & y_{2k} & y_{2m} & \\
P= & T3 & y_{31} & y_{32} & y_{3j} & y_{3k} & y_{3m} & \\
 & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
 & Ti & y_{i1} & y_{i2} & y_{ij} & y_{ik} & y_{im} & \\
 & \ldots & & & & & & \\
 & Tn & y_{n1} & y_{n2} & y_{nj} & y_{nk} & y_{nm} &
\end{array}
$$

Figure: 3 Pattern matrix

K. A Fast Algorithm For Mining Association Rules Based on Concept Lattice:

Yuan-Yuan Wang and et.al proposed a fast algorithm based on Concept lattice [17] for finding association rules from large and dynamic database. In this algorithm first, the building of concept lattice [18, 19] is done. Followed, by the mining of association rules using concept lattice is done. In this algorithm various Theorems shown in [17] are used to mine the frequent itemsets from concept lattice. The proposed algorithm finds all the frequent itemsets with only one scan of the database. Then after, finding all the frequent itemsets the process of generating association rules is same as for Apriori algorithm [1]. The experimental result of [17] proves that efficiency of proposed algorithm is better than Apriori algorithm in case of large and dynamic database.

L. Mining of association rules using frequent Itmset Lattice (20)::

Bay Vo, Bac Le presented a method to find the association rules by using itemset lattice [20]. The new method works in two steps. In first, it construct the itemset lattice which will represent the parent child relationship between frequent itemsets and in second step, the algorithm mines the association rules directly from the itemset lattice. The time taken to find association rules by using itemset lattice is very small as compare to previous algorithms. The process of building itemset lattice consumes more time but it is cover up by the time to find association rules. The itemset lattice also has the property of reuse i.e. if we want to mine association rules with many different minimum confidences in database which has the same minimum support; only one itemset is build for mining different association rules.

M. An efficient association rule mining algorithm Based on Coding and Constraints (21):

Association rule mining algorithm based on coding and constraint uses the properties of Apriori algorithm and makes some improvement based on it. The algorithm uses the sub-block coding method [21] for properties and applies constraints for antecedent and consequent [21] of the rules. In this method the attribute value is divided into decision attributes and non decision attributes [21]. Decision attribute appears in the antecedent of the association rule and Non-Decision attributes can only appear in the consequent of the rule. The result of this paper has shown that the new method reduces the number of candidate itemset generated and also reduces the number of times the database is scanned.

N. Hmfs Method:

Don-Lin Yang and et.al gives an improved and efficient Hash-based method called HMFS for finding the maximal frequent itemset [22]. The HMFS method combines the advantages of both Direct Hashing and Pruning (DHP) [23] and the Pincer-Search algorithm [24]. HMFS uses the hash technique of DHP algorithm to filter infrequent itemsets in bottom-up direction and uses top-down technique that is similar to the Pincer-Search algorithm but differ in the way to initialize the set of maximal frequent candidate itemsets.

The HMFS algorithm is more efficient than direct hash based method as the number of times the database scanned is greatly reduced and the process of finding maximal frequent itemset is also fast. Thus, the HMFS algorithm performs better than DHP and Pincer-Search algorithm.

O. Vector Based Method:

Zhi Lin, Guoming Sang, Mingyu Lu proposed a vector operation based method [25] for finding association rules. The proposed algorithm finds the association rule more efficiently and requires only one database scan to find all the frequent itemsets.

The process of generating frequent itemsets of this method consists of two steps:

a. Generation of Boolean matrix, where Boolean matrix is given as

$$M_{ij} = \begin{cases} 1 & I_j \text{ belongs to } T_i \\ 0 & I_j \text{ not belongs to } T_i \end{cases}$$

b. Then V-Apriori Algorithm [25] is applied to find the frequent itemset from Boolean matrix.

The Pseudo code for generation of frequent itemset is as follows:

a) Create an m*n Boolean matrix M according to def. 1 of [25]
b) Generate frequent 1-itemset in terms of def. 2 of [27]. Sort the itemset in descending value of their support count.
c) Generate frequent 2-itemset in terms of def. 3 [25] and save the result.
d) Use (k-1)-itemset to produce k-itemset.
e) Repeat the above steps until no more frequent k-itemset exist.
f) End;

The V-Apriori algorithm improves the performance of Apriori algorithm i.e. it overcomes both two problems of Apriori algorithm. The number of times the database scanned is greatly reduced and candidate itemset generated are also reduced. More ever, computation of matrix is simple as compare to perform join and prune operation as in case of classical Apriori algorithm.

V. COMPARISION OF VARIOUS METHODS

The comparison of various methods is shown below in Figure 4.

Method	Concept	Number of times database scanned	Number of candidate generated	Advantages
Apriori	Apply join and pruning operation along with Apriori property	Large	Large	1. Simple 2.one of the classical Algorithm to find association rules
FP-Growth	Construct FP-tree and then mine the frequent patterns from it.	Reduced to two	No	1. More efficient than Apriori. 2. No need to perform join and prune operation.

Hashing	Use Hash based technique	Reduced	Less	1. Improvement over traditional Apriori. 2. More time efficient 3. Easy to mine frequent patterns with the help of hash tables.
RAAT	Reduced Apriori Algorithm with Tag. Concept of tagging is used to speed up the process.	Reduced as compare to Traditional Apriori	Less as compare to Traditional Apriori	1. Reduces one redundant pruning operation of C2. 2. Saves time and increases efficiency.
QFP	Construct QFP-tree and then mines the frequent patterns from it.	Reduced to one	No	1. No need to sort the data items before making QFP-tree.
APFT	Combines Apriori algorithm and FP-tree structure of FP-Growth algorithm. Construct FP-tree and then use Apriori algorithm to mine FP-tree.	Reduced	Less	1. Does not generate the conditional pattern base and sub-conditional pattern. 2. Work faster than Apriori and FP-Growth Algorithm.
FP-Split	Generate equivalence class and then sort equivalence class in descending order to construct FP-Split tree.	Reduced to one	No	1. Improvement over FP-growth algorithm. 2. Construction of FP-Split tree consumes less time. 3. Efficient and scalable. 4. No filtering and sorting of items is required. 5. Header table and links are not scanned again and again while designing new node in the FP-Split tree.
MFP	Construct MFP-tree and then mines the frequent patterns from it directly.	Reduced to one	No	1. More efficient than FP-Growth. 2. Can be applied to any situation where FP-Growth or Apriori are suitable.
P_Matrix	Construct a binary pattern matrix and then perform AND operation on Boolean matrix rows to generate frequent patterns.	Reduced to one.	No	1. Reduces the temporal complexity and spatial complexity. 2. More efficient than Apriori algorithm.
Concept Lattice	Build the concept lattice, mines the frequent patterns directly from it.	Reduced as compare to FP-Growth and Apriori	Less as compare to FP-Growth and Apriori	1. Total time to build the concept lattice and finding frequent itemsets is shorter than that of Apriori. 2. Mostly used for finding frequent itemsets in case of large and dynamic database.
Frequent Itemset Lattice	Construct the frequent itemset lattice and then mine the association rules from it.			1. Frequent itemset lattice has the property of reuse. 2. Saves lot of time for mining of association rules.
Coding and Constraint	Sub-block coding method is used for properties and the constraints are made for the antecedent and consequent of rules.	Scanning size of database is reduced.	Less	1. Improves the operating efficiency. 2. More efficient than traditional approach. 3. Algorithm is simple and easy to maintain.
HMFS	Combines the advantages of both DHP and Pincer-Search algorithms.	Reduced database scan	Can filter out infrequent candidate itemsets.	1. Use the filtered candidate itemsets to find the maximal frequent itemsets. 2. Reduces the search space. 3. Better performance than DHP and Pincer-Search algorithms.

Vector Operation based Method	Construct the Boolean matrix and then finds the frequent itemsets via vector computation on matrix.	Requires only one database scan to generate Boolean matrix.	Less as frequent itemsets are find out through the AND operation on the vectors in the Boolean matrix.	1. Boolean matrix is stored in bit mode, So the memory space is greatly reduced. 2. As compare to the traditional Apriori algorithm the V-Apriori algorithm is improved in both time and space complexity.

Figure: 4 Comparision of various methods

VI. CONCLUSION

With the large database the process of finding association rules become difficult. Efficient methods are required to find association rules more quickly and efficiently. Different researchers are working on association rule mining to develop new methods. So, in this paper review on various known methods and some of recently developed methods has been presented. It is found that still a lot of work is required to be done to find out association rule in case of very large database and in situation where database is changing dynamically.

VII. REFERENCES

[1] Jawei Han and Michelline Kamber, "Data mining Concepts and Techniques", Morgan kaufman academic press, (2001).

[2] R.Agrawal and R.Srikant, "Fast Algorithm for Mining Association Rules", Proc. of the Int. Conf on Very Large Database, 1994, pp. 487- 499.

[3] Wei Zhang, Hongzhi Liao and Na Zhao, "Research on The FP-Growth Algorithm About Association Rule Mining", Proc. of Int. seminar on Business and Information Management, 2008, pp. 315-318.

[4] J.Han, J. Pei and Y. Yin,"Mining Frequent Patterns Without Candidate Generation". Proc. Of the ACM SIGMOD Int. Conf. on Management of Data, 2000, pp.1-12.

[5] J.Han, J.Pei, Y. Yin and R.Mao, "Mining Frequent Patterns without Candidate Generation: A Frequent Pattern Tree Approach", In Data mining and Knowledge Discovery, 2004, vol 8, pp.53-87.

[6] Changsheng Zhang and Jing Raun, "A Modified Apriori Algorithm with its application in Instituting Cross-Selling strategies of the Retail Industry", Poc. Of IEEE Int. conf. on Electronic Commerec and Business Intelligence, 2009, pp.515-518.

[7] Wanjun Yu, Xiachun Wang and Fangyi Wang, Erkang Wang, Bowen Chen, "The Research of Improved Apriori Algorithm for Mining Association Rules", Proc. of IEEE 11th Int. Conf. on communication Technology, 2008, pp. 513-516.

[8] Liu Jing, Qiu chu and et.al, "An Improved Algorithm for Early Warning of Equipment Failure", IEEE Int. Conf.,2009, pp. 450-452.

[9] Dongme Sun and et.al, "An Algorithm to Improve the Effectiveness of Apriori Algorithm", Proc. of 6th IEEE Int. Conf. on Cognitive Informatics", 2007, pp. 385-390.

[10] Song Hai- Shang,"A Fast Algorithm for Mining Maximum Frequent Itemsets", Int. conf. on computer Application and Research, 2004, pp. 45-46.

[11] Gang Yang, Hong Zhoa and et.al, "An implementation of Improved Apriori Algorithm", proc. of the IEEE 8th Int. Conf. on Machine Learning and Cybernetics, 2009, pp. 1565-1569.

[12] Li juan, Ming De-ting,"Research of association Rule Mining Algorithm Based on FP-Tree", IEEE, 2010, pp. 559-563.

[13] Qihua Lan, Defu Zhang, Bo Wu,"a New Algorithm for Frequent itemsets Mining", IEEE Global congress on intelligent system, 2009, pp. 360-364.

[14] Chin-Feng Lee and Tsung-Hsien Shen,"An FP-Split Method for Fast Association Rule Mining", IEEE, 2005, pp. 459-463.

[15] Jun Gao, "A New Association Rule Mining algorithm and Its Application", IEEE 3rd Int. Conf. on Advanced Computer Theory and Engineering (ICACTE), 2010, vol 5, pp. 122-125.

[16] Sixue Bai, Xinxi Dai, "An Efficiency Apriori Algorithm: P_Matrix Algorithm", IEEE First Int. Symposium on Data, Privacy and E-Commerce, 2007, pp. 101-103.

[17] Yuan- Yuan Wang, Xue-Gang Hu, "A Fast Algorithm for Mining of Association Rules Based on concept Lattice", Proc. of 3rd Int. Conf. on Machine Learning and Cybernetics,2004, pp. 1687-1691.

[18] Godin, R.Missaui, R.,and Alaoui, H., "Learning Algorithms Using a Galosis Lattices Structure", Proc. 3rd Int. Conf. on Tools for artificial Intelligence, 1991, San Jose, Calif, IEEE Computer Society Press, pp. 22-29.

[19] Robert Godin, Rokia Missaui, Hassan Alaoui, "Incremental Concept formation Algorithm Based on GAlosis Concept Lattice", In computational Intelligence, 1995, pp. 246-267.

[20] Bay Vo, Bac Le, "Mining Traditional Association Rules Using Frequent Itemsets Lattice", IEEE, 2009, pp. 1401-1406.

[21] Zhi Liu, Mingyu Lu and et.al, "An Efficient Association Rules Mining Algorithm Based on Coding and Constraints", 2009

[22] Don- Lin Yang, Ching-Ting Pan and et.al, "An Efficient Hash Based Method for Discovering the Maximal Frequent set", IEEE, 2001, pp. 511-516.

[23] J.S Park, M. S. Chen and P. S Yu, "An Efficient Hash Based Algorithhm for Mining Association Rules", Proc. of the ACM SIGMOD, 1995, pp. 175-186.

[24] D. Lin and Z. M. Kedem, "Pincer –Search: A New Algorithm for Discovering the Maximal Frequent Set",

Proc. of 6th Int. Conf. on extending database Technology, 1998.

[25] Zhi Liu, Guoming Sang, Mingyu Lu, "A Vector Operation Based Fast Association Rules Mining Algorithm", Proc. of Int. Joint Conf. on Bioinformatics, System Biology and Intelligent Computing, 2009, pp. 561-564.