BIJIT - BVICAM's International Journal of Information Technology
Bharati Vidyapeeth's Institute of Computer Applications and Management (BVICAM), New Delhi (INDIA)

A Novel Pruning Approach for Association Rule Mining

Lalit Mohan Goyal[1], M. M. Sufyan Beg[2] and **Tanvir Ahmad[3]**

Submitted in April, 2014; Accepted in December, 2014

Abstract – The problem of Association rule mining (ARM) can be solved by using Apriori algorithm consisting of 3-steps -Joining, Pruning and Verification. Pruning step plays an important role in eliminating weak candidate itemsets. In this paper, a new pruning step is proposed as an alternate to Apriori's pruning step. This alternative is depicted as a filtration step. Five experiments are carried out to claim that proposed pruning method also works as efficient as Apriori's pruning method.

Index Terms – Data mining, ARM (Association Rule Mining), Apriori algorithm, pruning.

1.0 INTRODUCTION

Data mining [8] is a method of extracting non-trivial, inherent, unfamiliar and practical information from large repositories. Association rule mining extracts frequent patterns [1], correlations [19], subsequences [12], substructures [11, 26, 27] or associations [1] among sets of items of the databases. An association rule is an implication $X \Rightarrow Y$, where X and Y are sets of items. The meaning of such expression is that transactions of a database which contain X likely to contain Y. For example- 95% of the students who buy a lap-top and a book related to computers can also purchase a pen-drive.

The rules thus discovered from the databases can be used to rearrange the related items together or can be used to make new market strategies which further increase the sales. Application domain of association rule mining is not only limited to the context of retail application but can also be used in the decision logics to the medical applications [7, 13].

This paper is organized as follows. Section 2. A describes the problem statement. Section 3 summarizes the related work in this field. In section 4, a modified *Apriori* algorithm is proposed with experimental results and in the last section, conclusion and future directions are stated followed by section of references.

2.0 PROBLEM STATEMENT

The following is a formal statement of the association rule mining problem to be solved. Let $N = \{i_1, i_2, \dots, i_m\}$ be a set containing *m* items. A set of items $X \subseteq N$ is called an itemset and an itemset having *k* numbers of items is known as *k*-itemset. Let $D = \{T_1, T_2, \dots, T_n\}$ be a set containing *n* transactions, where each transaction $T_j, 1 \le j \le n,$ is a set ofitems such that $T_j \subseteq N$. There is a unique identifier, *TID*, related with each transaction. We can say that a transaction T_j contains an itemset X if $X \subseteq T_j$. A *k*-itemset is said to be frequent only if all its *k* items are in some minimum number of transactions. An association rule is an implication of the form $X \Rightarrow Y$, where $X \subset N, Y \subset N$and $X \cap Y = \phi$. X is called antecedent and *Y* is called consequent; the rule means *X* implies *Y*. For a given set of transactions *D*, the problem of mining association rules is to generate all association rules that have certain user-defined minimum support, called *minsup*, and user defined minimum confidence, called *minconf*. Support of an association rule is defined as the ratio of the number of transactions that contain all the items of the set $X \cup Y$ to the total number of transactions in the database *D*, *i.e.*,$support = |\{T_j \epsilon D | X\ U\ Y \subseteq T_j\}|/|D|$. Confidence of an association rule is defined as the ratio of the number of transactions that contain all the items of the set $X \cup Y$ to the total number of transactions that contain all the items of the set *X*, *i.e.*,$confidence = |\{T_j \epsilon D | X\ U\ Y \subseteq T_j\}|/|\{T_j \epsilon D | X \subseteq T_j\}|$. The problem of mining association rules is a two phase process. In first phase, all sets of items which occur with a frequency greater than or equal to the given minimum support are identified and in the second phase, all sets of rules that satisfy given minimum confidence are generated. All sets of items which satisfy minimum support are known as frequent itemsets and all sets of rules which satisfy minimum confidence are known as association rules. Here, the major cost of mining association rules is contributed by first phase only. It is for the reason that most of the researchers focused their investigations on identifying frequent itemsets.

To identify frequent itemsets Agarwal and Srikant [2] proposed an algorithm, called ***Apriori***. It is a three steps process: joining, pruning and verification as shown in Table 1.1. In the first step of this algorithm, the *k*-frequent candidate itemsets are generated by using joining operation which is defined on two frequent itemsets as if $\{x_1, x_2, \dots, x_{k-2}, x_{k-1}\}$ and $\{x_1, x_2, \dots, x_{k-2}, y_{k-1}\}$ are two *k*-1 frequent itemsets and $x_{k-1} < y_{k-1}$ then joining operation will yield the following *k*-frequent candidate itemset $\{x_1, x_2, \dots, x_{k-2}, x_{k-1}, y_{k-1}\}$. In the second step, these *k*-frequent candidate itemsets are pruned to generate potential *k*-frequent itemsets whose all subsets containing *k*-1 items are frequent. In the third step, potential *k*-frequent itemsets are verified by scanning all transactions of database *D* for a given minimum support. These steps are repeated until large *k*-frequent itemsets are generated. Here, large *k*-frequent itemsets mean that value of *k* should be as large as possible. *Apriori* algorithm uses bottom-up approach to generate *k*-frequent itemsets, so, initially, it requires 1-frequent itemsets

[2,] *Department of Computer Engineering, Aligarh Muslim University (AMU), Aligarh (UP), INDIA.*
[1, 3] *Department of Computer Engineering, Jamia Millia Islamia (A Central University), New Delhi, INDIA.*
E-mail: [1]*lalitgoyal78@rediffmail.com,*
[2]*mmsbeg@cs.berkeley.edu and* [3]*tahmad2@jmi.ac.in*

which can be generated by scanning all transactions of database *D* against all items of set *N* for a given minimum support.
Prerequisite for the *Apriori* algorithm are: A set of transactions (*D*), minimum support (*minsup*) and 1-frequent itemsets (F_1).

3.0 RELATED WORK

The very first paper which directly addresses the problem of association rule mining was given by Agarwal *et al* [1]. Their research motivates the direction of enriching the database with more functionality to process those queries which can increase the sales of any retail market. In this paper an algorithm known as *AIS (Agrawal, Imielinski, Swami)* was proposed to answer those queries. In *AIS* algorithm potential candidate itemsets are generated and verified simultaneously during database scan. Drawback of this algorithm is that all the association rules are generated with only one item in their consequent, *i.e.*, if a rule $X \Rightarrow Y$ is generated then $|Y| = 1$.
Agarwal and Srikant [2] proposed a popular algorithm, called ***Apriori***, which came out as revolution in this field. In this algorithm generation and verification steps are separated in two steps one after the other. Afterwards, research has been carried out to improve or extend the *Apriori* algorithm. The pruning method used in this algorithm is also used by Mannila *et al* [14].
An algorithm known as *DHP* used direct hashing and pruning technique which improved the *Apriori* significantly [18]. Alternately, Mueller [16] introduced prefix tree instead of hash tree.
A sampling approach was used by Toivonen [23]. The idea is to pick a random sample to determine the negative boundary that separate the large frequent itemsets from the small frequent itemsets and validate the results with the rest of database. Algorithm thus produces exact association rules in one full pass over the database. But if sample misinterprets the negative border then the whole process needs to be repeated.
Savasere *et al* [20] partitioned the database and generated all association rules by scanning the database two times only. Mueller [16] also used partition technique to generate the frequent itemsets.
A new method of overlapping generation and verification step is evolved by Brin *et al* [5]. In this method, Association rules are known as implication rules. These rules are based upon conviction instead of confidence. Here, conviction neither talks about the co-relation nor talks about the co-existence. Conviction gives equal importance to both antecedent and consequent of the rule. Moreover, it is unambiguous and measures actual implication. Because of these two features, implication rules are more interesting than association rules. Ahmed *et al* [4] elaborated and extended the direction of Brin *et al* [5].
Srikant and Agarwal [21] have carried out their research on interval data to generate quantitative association rules by measuring the value of each individual attributes greater than some expected value. Quantitative association rule can be like- 95% of the students who buy a lap-top and a book related to computers can also buy at least two pen-drives. But methodology of Srikant and Agarwal gets fail when applied to interval data where separation between data values has some meaning [15].

Table 1: Apriori Algorithm

Apriori (*D*, *minsup*, F_1)
// *D* is a set of Transactions, *minsup* is the given //minimum support, $F_1$is the collection of 1-frequent itemset.
1. $k = 2$;
//**Step 1: (Joining):**C_k is the collection of *k*-frequent candidate itemsets.
2. $\boldsymbol{while}(F_{k-1} \neq \phi)$
3. $\boldsymbol{do} C_k = \phi$;
4. $\boldsymbol{for}$ $(each\ pair\ of\ itemset\ of\ type\ \{x_1, x_2, \dots, x_{k-2}, x_{k-1}\} \epsilon\ F_{k-1} and \{x_1, x_2, \dots, x_{k-2}, y_{k-1}$
5. $\boldsymbol{do\ if}$ $(x_{k-1} < y_{k-1})$
//Z is the new itemset of size *k*.
6. $\boldsymbol{then}\ Z = \{x_1, x_2, \dots, x_{k-2}, x_{k-1}, y_{k-1}\}$;
7. $C_k = C_k \cup Z$;
//**Step 2: (Pruning)** -P_k is the collection of *k*-frequent potential itemsets.
8. $P_k = \phi$;
9. $\boldsymbol{for}$ $(each\ candidate\ itemset\ Z \epsilon\ C_k)$
10. $\boldsymbol{do\ if}$ ($each\ subset\ of\ Z containing\ k-1\ items \epsilon\ F_{k-1}$)
11. $\boldsymbol{then} P_k = P_k \cup Z$;
//**Step 3:**(**Verification**) - $F_k$is the collection of *k*- //frequent itemset.
12. $F_k = \phi$;
13. $\boldsymbol{for}$ $(each\ potential\ itemset\ Z \epsilon\ P_k)$
//$support()$ functions returns ratio of the number of transactions that contain all the items of the set *Z* to the total number of transactions in the database *D*.
14. $\boldsymbol{do\ if}$ ($support(Z) >= minsup$)
15. $\boldsymbol{then} F_k = F_k \cup Z$;
//Value of *k* is increased to find large frequent itemsets.
16. $k = k + 1$;
17. $\boldsymbol{Print} \cup_{i=1}^{k-2} F_i$;

Agrawal and Srikant [3] also focused on mining the sequential association rules which is a kind of navigation or noticing a generalized behavior followed. Here, rules can be like- 95% of the students who are buying a laptop; followed by buying a book related to computers; and followed by buying a pen drive it is observed that they can purchase an external mouse also.
Continuous association rule are those kinds of rules which are generated online. Hidber [10] investigated continuous association rules and provided flexibility to change given minimum support during first scan of database.
Due to update in database it may be possible that the large frequent itemsets may become small frequent itemsets and vice-versa. Therefore, association rules are kept maintained instead of generating a new set of rules [6].

Interesting information means knowing something unknown, covering a large portion of database, and potentially useful. Association rules generated at high levels are not interesting and association rules generated at low levels are not useful. Therefore, in order to get interesting and useful associations, multilevel association rules are generated by [9, 17, 22].

Concept of recursive median is used by [24, 25] which is a probabilistic approach to discover frequent itemsets. It does not consider all candidate itemsets with equal probability to be frequent itemsets.

Next section 4.1 and 4.2 demonstrates an example to explain the work carried out in this paper. Section 4.3 proposes the modified algorithm and section 4.4 shows the experimental results.

4.0 MODIFICATION OF APRIORI ALGORITHM

The modification applied to *Apriori* algorithm is exposed by taking following example.

4.1Apriori algorithm with example:

Let $N = \{1,2,3,\dots,10\}$ be a set of ten different types of items a customer can purchase. Let $D = \{T_1, T_2, T_3, .., T_{20}\}$ be a set of twenty independent transactions. Each transaction $T_j$have a transaction identifier, *TID*, and list of items purchased shown in Table 2 and let us fix user-defined minimum support, *i.e.*, *minsup*= 5. Frequency of each individual item purchased in database D is shown in Table 1.3.

It is easy to generate following 1-frequent itemsets $F_1 = \begin{pmatrix}\{1\},\{3\},\{4\},\{5\},\{6\},\\ \{7\},\{8\},\{9\},\{10\}\end{pmatrix}$by comparing the frequency of each item of Table 1.3 with the given minimum support. These itemsets have frequency greater than or equal to given minimum support.

Thereafter, *Apriori* algorithm's first step is executed and following 2-frequent candidate itemsets are generated.

$$C_2 = \begin{pmatrix}\{1,3\},\{1,4\},\{1,5\},\{1,6\},\{1,7\},\{1,8\},\{1,9\},\{1,10\},\{3,4\},\{3,5\},\\ \{3,6\},\{3,7\},\{3,8\},\{3,9\},\{3,10\},\{4,5\},\{4,6\},\{4,7\},\{4,8\},\{4,9\},\\ \{4,10\},\{5,6\},\{5,7\},\{5,8\},\{5,9\},\{5,10\},\{6,7\},\{6,8\},\{6,9\},\\ \{6,10\},\{7,8\},\{7,9\},\{7,10\},\{8,9\},\{8,10\},\{9,10\}\end{pmatrix}$$

In the second step, pruning is applied on the outcome of first step and following potential 2-frequent itemsets are generated.

$$P_2 = \begin{pmatrix}\{1,3\},\{1,4\},\{1,5\},\{1,6\},\{1,7\},\{1,8\},\{1,9\},\{1,10\},\{3,4\},\{3,5\},\\ \{3,6\},\{3,7\},\{3,8\},\{3,9\},\{3,10\},\{4,5\},\{4,6\},\{4,7\},\{4,8\},\{4,9\},\\ \{4,10\},\{5,6\},\{5,7\},\{5,8\},\{5,9\},\{5,10\},\{6,7\},\{6,8\},\{6,9\},\\ \{6,10\},\{7,8\},\{7,9\},\{7,10\},\{8,9\},\{8,10\},\{9,10\}\end{pmatrix}.$$

Table 2: Set of Transactions

Transaction	TID	Items Purchased	Transaction	TID	Items Purchased
T_1	101	{1,4,6,7, 8,9}	T_{11}	223	{3,7}
T_2	102	{1,3,4,5,6, 7,8,9}	T_{12}	507	{1,4,7, 8,9}
T_3	103	{3,4,6, 8,9}	T_{13}	345	{1,2,3,4,5, 9,10}
T_4	201	{1,4,6}	T_{14}	309	{1,4,6,7}
T_5	213	{1,5,7,8, 9,10}	T_{15}	316	{1,3,4,6, 8,9,10}
T_6	123	{3,4,6, 8,9}	T_{16}	224	{4,8,10}
T_7	205	{2,4,6,9}	T_{17}	508	{1,5,8,9}
T_8	234	{2,3,4,8}	T_{18}	346	{1,4,6,9,10}
T_9	301	{3,7,8}	T_{19}	356	{1,4,5,6,9}
T_{10}	306	{1,2,3,4, 6,7,9}	T_{20}	366	{1,4,6,9}

Table 3: Individual Items Frequencies

Item number	Frequency	Item number	Frequency
{1}	13	{6}	12
{2}	4	{7}	8
{3}	9	{8}	11
{4}	16	{9}	14
{5}	5	{10}	5

In third step, the frequencies of above itemsets present in the database *D*, shown in Table 4, are compared with given minimum support and following 2-frequent itemsets are generated.

$F_2 = \begin{pmatrix}\{1,4\},\{1,5\},\{1,6\},\{1,7\},\{1,8\},\{1,9\},\\ \{3,4\},\{3,6\},\{3,8\},\{3,9\},\{4,6\},\{4,7\},\{4,8\}\\ \{4,9\},\{5,9\},\{6,8\},\{6,9\},\{7,8\},\{7,9\},\{8,9\}\end{pmatrix}$. These itemsets are used to generate 3-frequent candidate itemsets in next iteration. All the above three steps are executed iteratively until the large *k*-frequent itemsets are generated. It is worth mentioning here that the following itemsets $(\{1,3\},\{3,7\},\{6,7\})$are not 2-frequent because frequency of these itemsets is less than the fixed minimum support as shown in Table 1.4.

Table 4: 2-Frequent Itemsets Frequencies

Potential 2-Frequent Itemsets	Frequency	Potential 2-Frequent Itemsets	Frequency
{1,3}	4	{4,8}	8
{1,4}	11	{4,9}	12
{1,5}	5	{4,10}	4
{1,6}	9	{5,6}	2
{1,7}	6	{5,7}	2
{1,8}	6	{5,8}	3
{1,9}	11	{5,9}	5
{1,10}	4	{5,10}	2
{3,4}	7	{6,7}	4
{3,5}	2	{6,8}	5
{3,6}	5	{6,9}	9
{3,7}	4	{6,10}	2
{3,8}	f6	{7,8}	5
{3,9}	6	{7,9}	5
{3,10}	2	{7,10}	1
{4,5}	2	{8,9}	8
{4,6}	12	{8,10}	3
{4,7}	5	{9,10}	4

In the second iteration, following 3-frequent candidate itemsets and following 3-frequent potential itemsets are generated by first and second step respectively.

$$C_3 = \begin{pmatrix} \{\mathbf{1},\mathbf{4},\mathbf{5}\},\{1,4,6\},\{1,4,7\},\{1,4,8\},\{1,4,9\},\{\mathbf{1},\mathbf{5},\mathbf{6}\}, \\ \{\mathbf{1},\mathbf{5},\mathbf{7}\},\{\mathbf{1},\mathbf{5},\mathbf{8}\},\{1,5,9\},\{\mathbf{1},\mathbf{6},\mathbf{7}\},\{1,6,8\},\{1,6,9\}, \\ \{1,7,8\},\{1,7,9\},\{1,8,9\},\{3,4,6\},\{3,4,8\} \\ \{3,4,9\},\{3,6,8\},\{3,6,9\},\{3,8,9\},\{\mathbf{4},\mathbf{6},\mathbf{7}\},\{4,6,8\}, \\ \{4,6,9\},\{4,7,8\},\{4,7,9\},\{4,8,9\},\{\ 6,8,9\},\{7,8,9\} \end{pmatrix}$$

$$P_3 = \begin{pmatrix} \{1,4,6\},\{1,4,7\},\{1,4,8\},\{1,4,9\},\{1,5,9\},\{1,6,8\}, \\ \{1,6,9\},\{1,7,8\},\{1,7,9\},\{1,8,9\},\{3,4,6\},\{3,4,8\}, \\ \{3,4,9\},\{3,6,8\},\{3,6,9\},\{3,8,9\},\{4,6,8\},\{4,6,9\}, \\ \{4,7,8\},\{4,7,9\},\{4,8,9\},\{\ 6,8,9\},\{7,8,9\} \end{pmatrix}$$

At this point, it is to be noted here that 3-frequent candidate itemsets$\{1,4,5\},\{1,5,6\},\{1,5,7\},\{1,5,8\},\{1,6,7\}$and $\{4,6,7\}$are bold faced. The reason for making itemsets bold faced will be discussed in part 3.2 of this section.

Frequency of the 3-frequent potential itemsets, shown in Table 1.5, is compared with given minimum support and following 3-frequent itemsets are generated.

$F_3 = \begin{pmatrix} \{1,4,6\},\{1,4,7\},\{1,4,9\},\{1,6,9\},\{1,7,9\}, \\ \{1,8,9\},\{3,4,6\},\{3,4,8\},\{3,4,9\},\{3,6,9\}, \\ \{4,6,8\},\{4,6,9\},\{4,8,9\},\{6,8,9\} \end{pmatrix}$.It is again worth mentioning here that following itemsets $\begin{pmatrix} \{1,4,8\},\{1,6,8\},\{1,7,8\},\{3,6,8\}, \\ \{3,8,9\},\{\ 4,7,8\},\{4,7,9\},\{7,8,9\} \end{pmatrix}$are not 3-frequent because frequency of these itemsets is less than minimum support. Similarly, in third iteration, following 4-frequent candidate itemsets$C_4 = \begin{pmatrix} \{\mathbf{1},\mathbf{4},\mathbf{6},\mathbf{7}\},\{1,4,6,9\}, \\ \{\mathbf{1},\mathbf{4},\mathbf{7},\mathbf{9}\},\{\mathbf{3},\mathbf{4},\mathbf{6},\mathbf{8}\}, \\ \{3,4,6,9\},\{\mathbf{3},\mathbf{4},\mathbf{8},\mathbf{9}\},\{4,6,8,9\} \end{pmatrix}$; followed by 4-frequent potential itemsets$P_4 = \begin{pmatrix} \{1,4,6,9\}, \\ \{3,4,6,9\},\{4,6,8,9\} \end{pmatrix}$; and then followed by 4-frequent itemsets $F_4 = \begin{pmatrix} \{1,4,6,9\}, \\ \{3,4,6,9\},\{4,6,8,9\} \end{pmatrix}$ are generated by first, second and third step respectively. At this point again, the significance of making the following 4-frequent candidate itemsets $\begin{pmatrix} \{\mathbf{1},\mathbf{4},\mathbf{6},\mathbf{7}\},\{\mathbf{1},\mathbf{4},\mathbf{7},\mathbf{9}\}, \\ \{\mathbf{3},\mathbf{4},\mathbf{6},\mathbf{8}\},\{\mathbf{3},\mathbf{4},\mathbf{8},\mathbf{9}\} \end{pmatrix}$ bold faced will be discussed in part 3.2 of this section. Frequency of 4-frequent potential itemsets is shown in Table 1.6. There will be no more 5-frequent candidate itemsets generated by *Apriori* algorithm for this example because none of two itemsets from following 4-frequent itemsets $\begin{pmatrix} \{1,4,6,9\},\{3,4,6,9\}, \\ \{4,6,8,9\} \end{pmatrix}$are possible to join using joining operation in next iteration and therefore, *Apriori* algorithm terminates.

4.2 Alternate to *apriori's* pruning step

In section3.1, it is observed that large *k*-frequent itemsets can be generated by modifying the second step of *apriori* algorithm. For this, not only *k*-frequent but *k*-infrequent itemsets are also generated in the third step of *Apriori* algorithm. Then, the output of the first step is filtered by taking help of all infrequent itemsets generated so far. It is obvious that this modification is applicable from second iteration onwards.

Continuing the discussion from part 3.1 of this section, in the first step of second iteration, bold faced 3-frequent candidate itemsets $\{1,6,7\}$ and $\{4,6,7\}$ and in the first step of third iteration, bold faced 4-frequent candidate itemset $\{1,4,6,7\}$ are useless to generate because itemset $\{6,7\}$ is not 2-frequent. Similarly, $\{1,4,5\},\{1,5,6\},\{1,5,7\},\{1,5,8\}$ are useless to generate because$\{4,5\},\{5,6\},\{5,7\},\{5,8\}$ are not 2-frequent.By eliminating
$\{1,4,5\},\{1,5,6\},\{1,5,7\},\{1,5,8\},\{1,6,7\}$and $\{4,6,7\}$itemsets
from 3-frequent candidate itemsets following 3-frequent potential itemsets are generated.

$$P_3 = \begin{pmatrix} \{1,4,6\},\{1,4,7\},\{1,4,8\},\{1,4,9\},\{1,6,8\},\{1,6,9\}, \\ \{1,7,8\},\{1,7,9\},\{1,8,9\},\{3,4,6\},\{3,4,8\},\{3,4,9\}, \\ \{3,6,8\},\{3,6,9\},\{3,8,9\},\{4,6,8\},\{4,6,9\},\{4,7,8\}, \\ \{4,7,9\},\{4,8,9\},\{6,8,9\},\{7,8,9\} \end{pmatrix}$$

Similarly, bold faced itemsets $(\{1,4,7,9\},\{3,4,6,8\},\{3,4,8,9\})$ from the 4-frequent candidate itemsets are useless to generate because 3-frequent itemsets$(\{3,6,8\},\{3,8,9\},\{4,7,9\})$ are not 3-frequent. This process will result the following set as a 4-frequent potential itemsets $P_4 = (\{1,4,6,9\},\{3,4,6,9\},\{4,6,8,9\})$. This method of pruning advocates an alternative to pruning step of *apriori* algorithm. In this paper, this process of eliminating some or all itemsets from *k*-frequent candidate itemsets, using infrequent itemsets generated, is expressed as a "filtration" for first step of *Apriori* algorithm.

Table 5: 3-Frequent Itemsets Frequencies

Potential 3-Frequent Itemsets	Frequency	Potential 3-Frequent Itemsets	Frequency
{1,4,6}	9	{3,4,9}	6
{1,4,7}	5	{3,6,8}	4
{1,4,8}	4	{3,6,9}	5
{1,4,9}	8	{3,8,9}	4
{1,5,9}	4	{4,6,8}	5
{1,6,8}	3	{4,6,9}	10
{1,6,9}	7	{4,7,8}	3
{1,7,8}	4	{4,7,9}	4
{1,7,9}	5	{4,8,9}	6
{1,8,9}	6	{6,8,9}	5
{3,4,6}	5	{7,8,9}	4
{3,4,8}	5		

Table 6: 4-Frequent Itemsets Frequencies

Potential 4-Frequent Itemsets	Frequency
{1,4,6,9}	5
{3,4,6,9}	5
{4,6,8,9}	5

4.3 Modified *apriori* algorithm

A modified *Apriori* algorithm is proposed in Table 7. Filtration step is an alternate to the pruning step used in *Apriori* algorithm. Prerequisite for modified *Apriori* algorithm are same as with the *Apriori* algorithm.

This new pruning method is based upon following lemma.

Lemma: if any itemset Z is infrequent then none of it superset can be frequent.

Proof: Let *Y* be an itemset containing all the items of set *Z* and number of items in set *Y* is more than number of items in set *Z*, *i.e.*, $Z \subset Y$. As if *Z* is infrequent, it can be stated that:-

$$Support(Z) < minsup \quad (1)$$

It is obvious that support of itemset *Y* can't be greater than support of any of its subset in a given database because cardinality of set *Y* is more than cardinality of set *Z*, *i.e.*, $|Z| < |Y|$. It can be said that:-

$$Support(Y) \leq Support(Z) \quad (2)$$

Using property of associativity from equation "(1)" and "(2)" it can be stated that:-

$$Support(Y) < minsup \quad (3)$$

In other words set *Y* can't be frequent if it is superset of any set *Z* which is infrequent.

Table 7: Modified *Apriori* Algorithm

Apriori_Filter (*D*, *minsup*, F_1)
// *D*is a set of Transactions, *minsup* is the given //minimum support, $F_1$is the collection of 1-//frequent itemset.
1. $k = 2$;
//Step 1: (Joining) - C_k is the collection of *k*-//frequent candidate itemsets.
2. $\boldsymbol{while}(F_{k-1} \neq \phi)$
3. $\boldsymbol{do} C_k = \phi$;
4. $\boldsymbol{for}$ $(each\ pair\ of\ itemset\ of\ type\ \{x_1, x_2, \dots, x_{k-2}, x_{k-1}\} \epsilon\ F_{k-1} and \{x_1, x_2, \dots, x_{k-2}, y_{k-1}\}$
5. $\boldsymbol{do\ if}$ $(x_{k-1} < y_{k-1})$
//Z is the new itemset of size *k*.
6. $\boldsymbol{then}$ $Z = \{x_1, x_2, \dots, x_{k-2}, x_{k-1}, y_{k-1}\}$;
7. $C_k = C_k \cup Z$;
//Step 2: (Filtration) -$P_k$is the collection of *k*-frequent potential itemsets and $IF_k$is the collection of *k*-infrequent itemset
8. $P_k = C_k$;
9. $IF_k = \phi$;
10. $\boldsymbol{for}$ $(each\ infrequent\ itemset\ Z\ \epsilon\ \bigcup_{i=2}^{k-1} IF_i)$
11. $\boldsymbol{do\ if}$ $(Z \subset \{Y | \exists Y \epsilon P_k\})$
12. $\boldsymbol{then}$ $P_k = P_k - Y$;
13. $IF_k = IF_k \cup Y$;
//Step 3:(Verification) -$F_k$is the collection of *k*-frequent itemset.$F_k = \phi$;
14. $\boldsymbol{for}$ $(each\ potential\ itemset\ Z \epsilon P_k)$
15. $\boldsymbol{do\ if}$ $(support(Z) \geq minsup)$
16. $\boldsymbol{then} F_k = F_k \cup Z$;
17. $\boldsymbol{else}$ $IF_k = IF_k \cup Z$;
//Value of *k* is increased to find large frequent itemsets.
18. $k = k + 1$;
19. $\boldsymbol{Print} \bigcup_{i=1}^{k-2} F_i$;

5.0 Experimental Results

Five experiments are performed on synthetic datasets generated as described by [2]. Each experiment is executed three times for both *Apriori* and proposed algorithm. Average time taken (in seconds) by both algorithm in each experiment is shown below in respect to minimum support value. Following parameters are set for each experiment.

1. Number of total transactions in database are 100000, *i.e.*, $|D| = 100K$, numbers of items are 100, *i.e.*, $|N| =$ 100,average size of transactions is 5, *i.e.*,$|T| = 5$, average size of maximal potentially large itemsets is 2,*i.e.*,$|I| = 2$, value of *correlation level* is set to 0.5, number of maximal potentially large itemsets are 200,*i.e.*,$|L| = 200$,and value of minimum support is changed from 4% to 10%with a step increment of 1%. Results of this experiment are plotted in Figure 1.
2. Number of total transactions in database are 100000, *i.e.*,$|D| = 100K$, numbers of items are 100, *i.e.*,$|N| = 100$, average size of transactions is 5, *i.e.*,$|T| = 10$, average size of maximal potentially large itemsets is 2, *i.e.*,$|I| = 2$, value of *correlation level* is set to 0.5, number of maximal potentially large itemsets are 200, *i.e.*,$|L| = 200$, and value of minimum support is changed from 10% to 20% with a step increment of 1%. Results of this experiment are plotted in Figure2.
3. Number of total transactions in database are 100000, *i.e.*,$|D| = 100K$, numbers of items are 100, *i.e.*,$|N| = 100$, average size of transactions is 5, *i.e.*,$|T| = 10$, average size of maximal potentially large itemsets is 4, *i.e.*,$|I| = 4$, value of *correlation level* is set to 0.5, number of maximal potentially large itemsets are 200, *i.e.*,$|L| = 200$, and value of minimum support is changed from 10% to 20% with a step increment of 1%. Results of this experiment are plotted in Figure3.
4. Number of total transactions in database are 100000, *i.e.*,$|D| = 100K$, numbers of items are 100, *i.e.*,$|N| = 100$, average size of transactions is 5, *i.e.*,$|T| = 20$, average size of maximal potentially large itemsets is 4, *i.e.*,$|I| = 4$, value of *correlation level* is set to 0.5, number of maximal potentially large itemsets are 200, *i.e.*,$|L| = 200$, and value of minimum support is changed from 15% to 30% with a step increment of 3%. Results of this experiment are plotted in Figure4.
5. Number of total transactions in database are 100000, *i.e.*,$|D| = 100K$, numbers of items are 100, *i.e.*,$|N| = 100$, average size of transactions is 5, *i.e.*,$|T| = 20$, average size of maximal potentially large itemsets is 6, *i.e.*,$|I| = 6$, value of *correlation level* is set to 0.5, number of maximal potentially large itemsets are 200, *i.e.*,$|L| = 200$, and value of minimum support is taken6%, 9%, 12%, 15%, 20%, and 25%. Results of this experiment are plotted in Figure5.

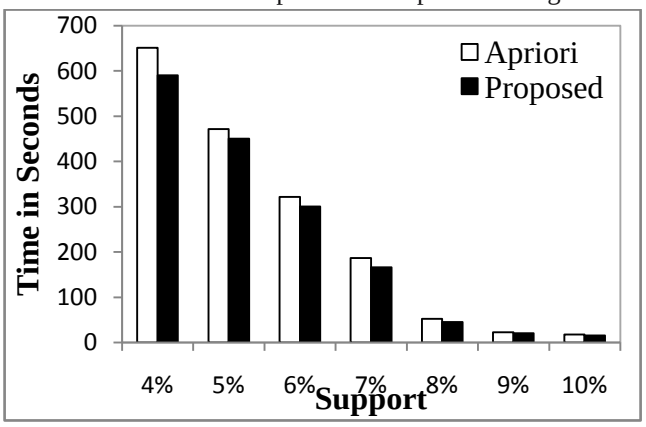

Figure 1: Apriori algorithm vs.Proposedalgorithm for $|D| = 100000,\quad |N| = 100,\quad |T| = 5,\quad |I| = 2.$

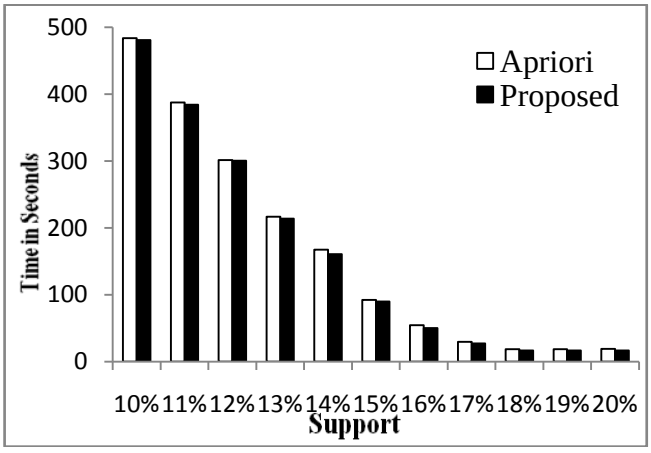

Figure 2: Apriori algorithm vs. Proposed algorithm for $|D| = 100000,\quad |N| = 100,\quad |T| = 10,\quad |I| = 2.$

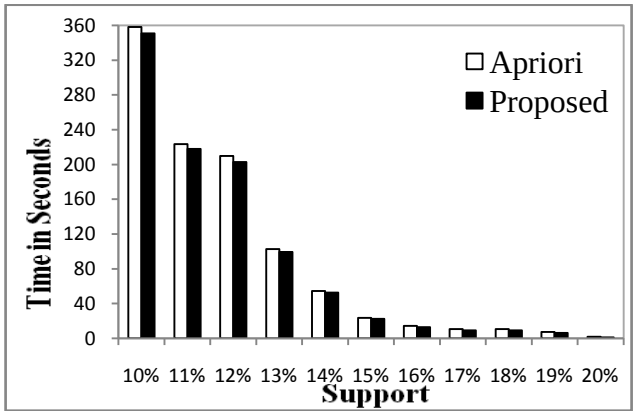

Figure 3 Apriori algorithm vs. Proposed algorithm for $|D| = 100000,\quad |N| = 100,\quad |T| = 10,\quad |I| = 4$

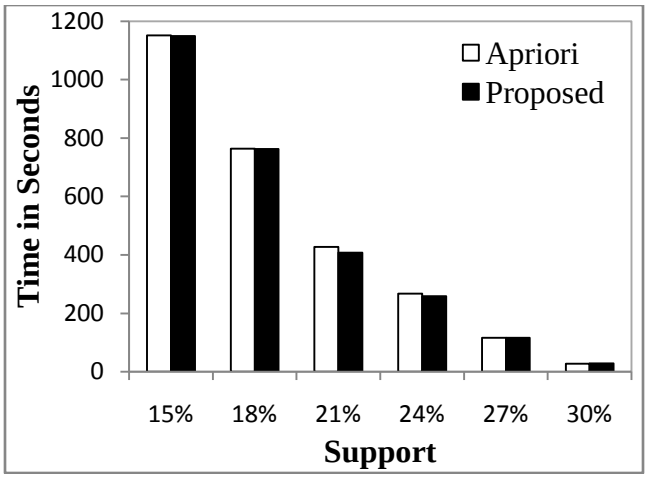

Figure 4: Apriori algorithm vs.Proposedalgorithm for

$|D| = 100000, \quad |N| = 100, \quad |T| = 20, \quad |I| = 4$

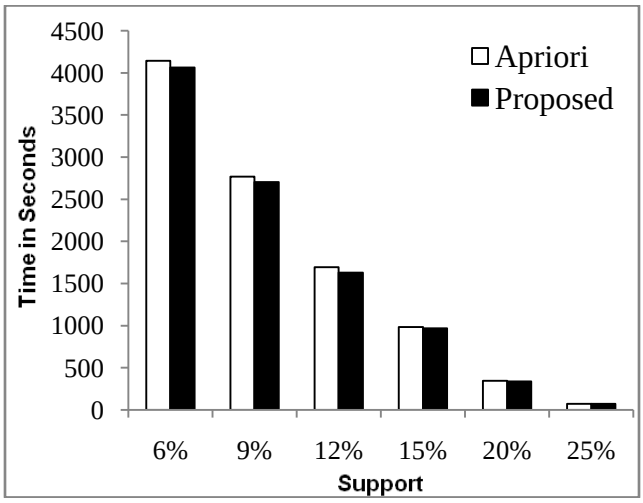

Figure 5 Apriori algorithm vs. Proposed algorithm for

$|D| = 100000, \quad |N| = 100, \quad |T| = 20, \quad |I| = 6$

All experimental results show that proposed algorithm improves *Apriori*algorithm.

6.0 CONCLUSION AND FUTURE WORK

In this paper, a new pruning method is proposed as an alternate to pruning method of *Apriori* algorithm. This method is expressed as a filtration for joining operation of *Apriori* algorithm. By using this new method, same candidate *k*-frequent itemsets are generated as by the Apriori's pruning method. It is observed that proposed approach works as efficient as the existing method. In future, proposed pruning approach may be helpful in *FP*-Growth, Eclat algorithm.

7.0 ACKNOWLEDGEMENT

Lalit Mohan Goyalis thankful to the management of Noida Institute of Engg. & Tech., Greater Noida for providing academic leaves.

8.0 REFERENCES

[1]. Agrawal R, Imielinski T, Swami A.,"Mining association rules between sets of items in large databases". In *Proceedings of the ACM-SIGMOD international conference on management of data (SIGMOD'93)*, Washington, DC, pp. 207–216, 1993.

[2]. Agrawal R, Srikant R.,"Fast algorithms for mining association rules". In *Proceedings of the international conference on very large data bases (VLDB'94)*, Santiago, Chile, pp. 487–499, 1994.

[3]. Agrawal R, Srikant R.,"Mining sequential patterns". In *Proceedings of the 1995 international conference on data engineering (ICDE'95)*, Taipei, Taiwan, pp. 3–14, 1995.

[4]. Ahmed K M, El-Makky NM, Taha Y., "A note on "beyond market basket: generalizing association rules to correlations"". SIGKDD Explorations 1: 46–48, 2000.

[5]. Brin S, Motwani R, Silverstein C.,"Beyond market basket: generalizing association rules to correlations", In *Proceeding of the ACM-SIGMOD international conference on management of data (SIGMOD'97)*, Tucson, AZ, pp. 265–276, 1997.

[6]. Cheung DW, Han J, Ng V, Wong CY.,"Maintenance of discovered association rules in large an incremental updating technique". In: *Proceeding of the 1996 international conference on data engineering (ICDE'96)*, New Orleans, LA, pp. 106–114, 1996.

[7]. Goyal, L.M., Beg, M.M.S., "An efficient filtration approach for mining association rules," *Computing for Sustainable Global Development (INDIACom - 2014), 2014 International Conference*, pp.178-185, 5-7 March 2014.

[8]. Goyal, L.M., Sufyan Beg, M.M., "Improved filtration step for mining association rules," *Data Mining and Intelligent Computing (ICDMIC), 2014 International Conference*, pp.1-4, 5-6 Sept. 2014.

[9]. Goyal, L. M.,Sufyan Beg, M.M., "Evaluation of filtration and pruning approach for Apriori algorithm," *Computer and Communication Technology (ICCCT), 2014 International Conference*, pp.23-28, 26-28 Sept. 2014.

[10]. Han J, Cheng H, Xin D, Yan X.,"Frequent pattern mining: current status and future directions". *Data Min. Knowl.Discov. (DATAMINE)* 15(1):55-86, 2007.

[11]. Han J, Kamber M. *Data mining: concepts and techniques*.2nd edn. Morgan Kaufmann.

[12]. Han J, Fu Y., "Discovery of multiple-level association rules from large databases", In *Proceeding of the international conference on very large data bases (VLDB'95)*, Zurich, Switzerland, pp. 420–431, 1995.

[13]. Hidber C.,"Online association rule mining". In *Proc. ACM SIGMOD Int. Conf. on Management of Data*, pages 145-154, Philadephia, Pennsylvania, USA, 1999.

[14]. InokuchiA,Washio T, Motoda H.,"An apriori-based algorithm for mining frequent substructures from graph data". In *Proceeding of the European symposium on the*

principle of data mining and knowledge discovery (PKDD'00), Lyon, France, pp. 13–23, 2000.

[15]. Ji X, Bailey J, Dong G.,"Mining minimal distinguishing subsequence patterns with gap constraints". In *Proceeding of the international conference on data mining (ICDM'05)*, Houston, TX, pp. 194–201, 2005.

[16]. M. M. S. Beg, C. P. Ravi Kumar,"Application of Parallel and Distributed Data Mining in e-Commerce", *J.IETE Technical Review special issue on e-Commerce*, vol. 17, no. 4, pp. 189-195, 2000.

[17]. Mannila H, Toivonen H, VerkamoAI.,"Efficient algorithms for discovering association rules". In: *Proceeding of the AAAI'94 workshop knowledge discovery in databases (KDD'94)*, Seattle,WA, pp. 181–192, 1994.

[18]. Miller RJ, Yang Y.,"Association rules over interval data",In*Proceeding of the ACMSIGMOD international conference on management of data (SIGMOD'97)*,Tucson, AZ, pp. 452–461, 1997.

[19]. Mueller A., "Fast sequential and parallel algorithms for association rule mining: A comparison",*Technical CS-TR-3515*, University of Maryland, College Park, August 1995, pp. 1-5, 1995.

[20]. N. Gangal, M. M. S. Beg, "Finding the Interesting Rules" First.*Proc. International Conference on Data Mining (DMIN'08)* - a track at the 2008 World Congress in Computer Science, Computer Engineering and Applied Computing (WORLDCOMP'08), Las Vegas, USA, July 14-17, 2008, pp. 689-694, 2008.

[21]. Park J S, Chen M S, Yu P S.,"An effective hash-based algorithm for mining association rules", In *Proceeding of the ACM-SIGMOD international conference on management of data (SIGMOD'95)*, San Jose, CA, pp. 175–186, 1995.

[22]. Piatetsky-Shapiro G.,"*Notes ofAAAI'91 workshop knowledge discovery in databases (KDD'91)"*, AAAI/MIT Press, Anaheim, CA, 1991.

[23]. Savasere A, Omiecinski E, Navathe S.,"An efficient algorithm for mining association rules in large databases", In *Proceeding of the 1995 international conference on very large data bases (VLDB'95)*, Zurich, Switzerland, pp. 432–443, 1995.

[24]. Srikant R, Agrawal R.,"Mining sequential patterns: generalizations and performance improvements",In: *Proceeding of the 5th international conference on extending database technology (EDBT'96)*, Avignon, France, pp. 3–17, 1996.

[25]. Srikant R, Agrawal R.,"Mining generalized association rules", In *Proceeding of the international conference on very large data bases (VLDB'95)*, Zurich, Switzerland, pp. 407–419, 1995.

[26]. Toivonen H.,"Sampling large databases for association rules", In *Proceeding of the international conference on very large data bases (VLDB'96)*, Bombay, India, pp. 134–145, 1996.

[27]. V. Sharma, M. M. S. Beg.,"A Probabilistic Approach to AprioriAlgorithm",*International Journal of Granular Computing, Rough Sets and Intelligent Systems (IJGCRSIS)*, Inderscience Publishers, ISSN (Online): 1757-2711 - ISSN (Print): 1757-2703, vol. 2, no. 3, 2012, pp. 225-243, 2012.

[28]. V. Sharma, M. M. S. Beg,"A Probabilistic Approach to Apriori Algorithm", *IEEE International Conference on Granular Computing (GrC 2010)*, Silicon Valley, USA, August 14-16, IEEE Computer Society Press, pp. 402-408, 2010.

[29]. Yan X, Zhou XJ, Han J.,"Mining closed relational graphs with connectivity constraints", In *Proceeding of the ACM SIGKDD international conference on knowledge discovery in databases (KDD'05)*, Chicago, IL, pp. 324–333, 2005.

[30]. Yan X, Zhu F, Han J, Yu PS., "Searching substructures with superimposed distance", In *Proceeding of the international conference on data engineering (ICDE'06)*, Atlanta, 2006.