I.J. Information Technology and Computer Science, **2017, 1, 58-64**
Published Online January 2017 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijitcs.2017.01.07

Enhancing the Performance in Generating Association Rules using Singleton Apriori

K.Mani
Department of Computer Science, Nehru Memorial College, Puthanampatti, Tiruchirappalli, 621 007, Tamilnadu, India
E-mail: nitishmanik@gmail.com

R.Akila
Department of Computer Science, Nehru Memorial College, Puthanampatti, Tiruchirappalli, 621 007, Tamilnadu, India
E-mail: grr.akila@gmail.com

***Abstract*—**Association rule mining aims to determine the relations among sets of items in transaction database and data repositories. It generates informative patterns from large databases. Apriori algorithm is a very popular algorithm in data mining for defining the relationships among itemsets. It generates 1, 2, 3,…., n-item candidate sets. Besides, it performs many scans on transactions to find the frequencies of itemsets which determine 1, 2, 3,…, n-item frequent sets. This paper aims to eradicate the generation of candidate itemsets so as to minimize the processing time, memory and the number of scans on the database. Since only those itemsets which occur in a transaction play a vital role in determining frequent itemset, the methodology that is proposed in this paper is extracting only single itemsets from each transaction, then 2,3,..., n itemsets are generated from them and their corresponding frequencies are also calculated. Further, each transaction is scanned only once and no candidate itemsets is generated both resulting in minimizing the memory space for storing the scanned itemsets and minimizing the processing time too. Based on the generated itemsets, association rules are generated using minimum support and confidence.

***Index Terms*—**Apriori, Candidate itemsets, Frequent itemsets, Minimum Support, Minimum Confidence, Single Scan.

I. INTRODUCTION

Data mining is a popular and an interesting domain for the reception of potential patterns [1] [10]. It is used for extracting or finding the knowledge from large amount of data [2] [11]. The term data mining is appropriately named as Knowledge Discovery in Databases. The discovered knowledge is used for prediction and analysis purposes of decision making patterns. Association rules are one of the most important enterprise data mining applications [6]. It deals with the data from multiple heterogeneous data sources and also extracting data in a single step from data sources which is time and space consuming [9]. Apriori algorithm is one of the most popular algorithms which use support and confidence [15] [17]. The *support* value of the itemset X with respect to the transaction T is defined as the proportion of transactions in the database which contains the itemset X and the *confidence* value of a rule, X => Y, with respect to a set of transactions T, is the proportion of the set of transactions that contains the itemset X which also contains the itemset Y [7] [16]. Thus, association rules are useful for discovering interesting relationships that are hidden in large datasets [3] [18].

Mining association rules using classical Apriori algorithm has several limitations viz., consideration of only the presence and absence of items, large number of scans on database and also the generation of candidate itemsets which may result in waste of time and space [5]. However, the generation of a large number of candidate itemsets and scanning the database several times individually face few challenges [7] [13], such as time utilization, expensiveness and inefficiency for very large databases. Several improved Apriori approaches viz., record filter, intersection, matrix based, set size frequency, interest item [4] [12] etc., have been proposed in literatures by considering transactions removal based on the number of itemsets, size of itemsets, usage of transaction identifiers, matrix construction for the presence and absence of items and considering only the interested itemsets. But, this paper is designed to eradicate the generation of candidate itemsets and to minimize the number of scans on the database. In the traditional Apriori, 1, 2, 3, …, n candidate itemsets are generated to find associations rules. The generated 1, 2, 3, …, n candidate itemsets must be stored in memory until all transactions are scanned for detecting the existence of 1, 2, 3,…, n candidate itemsets. After the generation of candidate itemsets, prune step scans the database to identify which candidate itemsets are frequent itemsets [16]. This leads to too many scans on database which is a memory and time consuming process [14].

The concept conceived in this paper is that an itemset plays a major role in determination of 1, 2, 3, …, n-itemsets only if it occurs in a transaction. Thus, the database is scanned only once for each transaction to extract only single itemsets. Single scan on the database also increases speed in execution. The remaining 2, 3, …, n-itemsets are generated from these single itemsets. This requires no separate candidate itemset generation. The generated single itemsets are stored only in buffer and also the number of single itemsets is always less than the number of candidate itemsets. Thus memory utilization is minimized. Besides, as no separate candidate itemset is generated and only the itemsets those occur in the transactions are taken into consideration, there is no prune step too, which leads to fast processing in terms of detection of frequent itemsets and speeds up the execution process.

The rest of the paper is organized as follows. The related work based on Singleton Apriori algorithm is presented in section 2. Section 3 focuses on the proposed methodology. Proposed work with an example is described in section 4. The results are discussed in section 5. Finally, section 6 ends with conclusion.

II. Related Work

In [2], Peng Peng, Qianli Ma and Chaoxiong Li proposed the application research of component technique in data mining and they implemented varied core algorithms of data mining in the form of components and used the component library to achieve the organization, management and retrieval of the components. Through componentization of data mining algorithm wrapping individual business process of data mining in the form of components, the efficiency and quality of data mining softwares are improved significantly to meet various application demands and thus the application of data mining technology can be broadened. Caixian Chen, Huijian Han and Zheng Liu [3] proposed an improved K-Nearest Neighbour (KNN) classification method based on Apriori algorithm to modify the determination method of nearest neighbor number k and reduces the time complexity of classification. The experiment demonstrated that the time complexity of the improved KNN question classification method was relatively smaller and the classification accuracy was high. In [4], Arti Rathod, Mr. Ajaysingh Dhabariya and Chintan Thacker stated that the association rules are efficient to reveal all the interesting relationships in a large database with huge amount of data. They revealed that apriori algorithm has several limitations like scanning time, memory optimization, candidate generation which can be solved by several improved Apriori approaches.

Zhi Liu, Tianhong Sun and Guoming Sang [5] described that Hash table technology can be effectively reduce frequent itemset size especially the frequent 2-itemsets and running time. Although the above said technology requires additional space to store the Hash table, the running time can be greatly reduced. Sadegh Bafandeh Imandoust and Mohammad Bolandraftar [6] discussed about different types of predictions which use data mining techniques viz., classification which predicts into what category or class a case falls and regression and KNN. They further explained KNN for classification, regression and the application of KNN in different fields.

In [7], Harpreet Singh and Renu Dhir described that apriori algorithm suffers from two limitations. They are the generation of large number of candidate itemsets and scanning the database too many times. Their proposed new method is based on transactional matrix and transaction reduction which solves both the problems. It helps in mining efficient frequent itemsets. In this method, it is easy to generate the frequent itemsets directly from the transactional matrix, if the transactional matrix is generated.

In [8], Sunitha Vanamala, L.Padma sree and S.Durga Bhavani discussed the rare association rule which refers to an association rule formed between frequent and rare items or among rare items and indicated that to mine interesting rare association rules, single minimum support approaches are not useful. Further, they proposed an algorithm based on multiple support apriori called MSApriori and uses vertical database format called MSApriori_VDB. The experimental results showed that the proposed algorithm outperformed in both memory requirement and execution time as the reduction of the number of database scans.

Jogi Suresh and T.Ramanjaneyulu [9], stated that finding frequent itemsets is not trivial because of its combinatorial explosion. They concluded with many algorithms that can further be used for finding informative patterns from the complex data sources. In [10], P. Ajith, B. Tejaswi and M.S.S.Sai used association rules instead of tree based classification. Association rules aims at discovering implicative tendencies among sets of items in transaction databases that can be valuable information for the decision maker those are absent in tree based classifications. Thus, they proposed a new interactive approach to prune and filter discovered rules. First, they proposed to integrate user knowledge in the post processing task. And secondly, proposed a rule schema formalism extending the specifications to obtain association rules from knowledge base. The results are better to understand and to be applied to real time use than tree based classifications.

Ziauddin, Shahid Kammal, Khaiuz Zaman Khan and Muhammad Ijaz Khan [11] discussed potential problems for association rule mining like generalization of algorithms, single scan, efficient, effective and scalable methods for association rule mining, database-independent measurements and mechanisms for deep understanding, interpretation of patterns and time and space consumption was reported by them. They further suggested some effective approaches are needed to overcome these problems and to decrease the time as well as space. In [12], Badri Patel, Vijay K Chaudhari, Rajneesh K Karan and YK Rana proposed an Ant Colony Optimization (ACO) algorithm which is a meta heuristic algorithm being inspired by the foraging behavior of ant colonies for optimizing the association rule generation using Apriori algorithm. It described a method for the problem of association rule mining.

Deepa S. Deshpande [13] proposed a new method of association rule mining using pattern generation. Generation of patterns, frequent itemset and derivation of association rules was performed as a three step process and a comparison between the new method and the traditional Apriori showed that the behavior of the proposed method is much more similar to Apriori algorithm with less memory space and reduction in multiple times scanning of database. In [14], Darshan M. Tank proposed an algorithm to improve Apriori algorithm by the way of a decrease of pruning operations, which generates the candidate 2-itemsets by the apriori-gen operation. Besides, it adopted the tag-counting method to calculate support quickly to overcome the bottleneck of poor efficiency of counting support. Padam Gulwani [15] proposed an approach that uses the idea of representative rules to prune the rules first using minimum support, confidence and then hides the sensitive rules. It was stated that the experimental results showed that proposed approach hides the more number of rules in minimum number of database scans compared to existing algorithms based on the same approach.

Sonia Setia and Dr. Jyoti [16] presented a survey on Multi-Level Association Rule Mining and mining algorithms. It was stated that main task of association rule mining technique is to find the frequent patterns by using minimum support thresholds decided by the user. This algorithm is inefficient because it scans the database many times. Second, if the database is large, it takes too much time to scan the database. For many cases, it is difficult to discover association rules among the objects at low levels of abstraction and stated that Apriori algorithm does not mine the data on multiple levels of abstraction.

Thabet Slimani and Amor Lazzez [17] provided a brief review and analysis of the current status of frequent pattern mining and discussed some promising research directions. Additionally, this paper included a comparative study between the performances of the described approaches. It was stated that the process of data mining produces various patterns from a given data source and the most recognized data mining tasks are the process of discovering frequent itemsets, frequent sequential patterns, frequent sequential rules and frequent association rules. An important progress was made to complex algorithms such as sequential pattern mining, structured pattern mining, correlation mining.

Manish Saggar, Ashish Kumar and Agrawal Abhimanyu Lad [18] applied some improvements in Genetic Algorithms (GAs) to help the rule based systems used for classification and described that in general the rule generated by Association Rule Mining technique do not consider the negative occurrences of attributes in them, but by using GAs over these rules the system can predict the rules which contains negative attributes. The main motivation for using GAs in the discovery of high-level prediction rules is that they perform a global search and cope better with attribute interaction than the greedy rule induction algorithms open used in data mining.

III. Proposed Work

This section describes various works related to Singleton Apriori. Apriori algorithm is one of the acceptable efficient techniques in data mining to establish relationships among data sets. Its efficacy can be enhanced by minimizing the large number of scans on database and eliminating the generation of candidate itemsets. Most of the researchers have mentioned that the traditional apriori algorithm needs more time to produce response, as it scans database many times. Also, it requires more memory space and processor time because of its requirement to generate candidate itemsets and to store them until all transactions are scanned.

As an enhancement, the proposed work aims to reduce the number of scans on the database and also to eliminate the generation of candidate itemsets. For that, it focuses on extracting only the itemsets which occur in the transactions, but not generating candidate itemsets. It also scans the database only once for each transaction to find 1, 2, 3,…,n-itemsets but not considering an itemset which does not occur in a transaction because it plays no role to determine its frequency. It is achieved by performing scan on the database to extract only single itemsets. The generated single itemsets are then used to generate other 2, 3,…, n-itemsets and their frequencies are also calculated based on their existence in the transactions. Thus, the total number of retrievals depends on the retrievals of single itemsets with n transactions and k items are calculated. This proposed methodology consists of the following four phases and it is also shown in Algorithm 1.

1) *Single Itemset Generation*

Let there are n number of transactions, k number of itemsets. To extract 1-itemsets, each transaction of database is scanned only once and the extracted single itemsets are stored in subsets of single itemsets (SS_s) (i.e) and also $SS_s \subseteq T_t$. It is calculated as

$$SSs = \begin{cases} \{Ii\}, \text{if Ij} \in \text{Tt} \\ \emptyset, \quad \text{if Ij} \notin \text{Tt} \end{cases} \tag{1}$$

where $1 \leq s \leq k$, $1 \leq j \leq k$ and $1 \leq t \leq n$.

2) *Join*

To find 2 to n-itemsets, subsets are generated from 1itemsets obtained from phase 1 for each transaction using (2)

$$\text{PS(k, m)} = \sum_{m=2}^{k} \binom{k}{m} = 2^k - \sum_{m=0}^{1} \binom{k}{m} \tag{2}$$

where k is number of items in SS_i, $1 \leq i \leq p$,
m is the type of itemsets (2, 3, 4,…,k) and
2^k is the total number of subsets of itemsets.

3) *Frequency Calculation*

The number of times the itemsets occur in each transaction is calculated using (3)

$$FT1Ij = \begin{cases} 1, if\ Ii \in T1 \\ 0, if\ Ij \notin T1 \end{cases} \tag{3}$$

where F_{T1Ij} is the number of occurrences of itemsets I_j and

1 ≤ j ≤k in the transaction T_1. Similarly, the number of occurrences of itemsets I_j, 1≤j≤k for the other transactions T_i, 2≤i≤n and and 1≤s≤k is calculated using (4)

$$FTiIj = \begin{cases} FTiIj + 1, \text{if Ij} \in \text{Ti and Ij} \in \text{SSs} \\ 1, \quad if\ Ij \in Ti\ and\ Ij \notin S \\ 0, \quad if\ Ij \notin Ti \end{cases} \tag{4}$$

The total number of retrievals for single itemsets with n transactions and k items is calculated using (5)

$$TNRnk = \sum_{i=1}^{n} \sum_{j=1}^{k} Sij \tag{5}$$

where

$$Sij = \begin{cases} 1, & if\ Ij \in Ti \\ 0, & otherwise \end{cases}$$

4) *Association Rule Generation*

Based on the minimum support and confidence, association rules are generated.

Algorithm1 SingletonApriori*(TransDB,Min_Supp,Min_conf)*

```
//Ti : ith transaction
//SIS : Store single itemsets
//ND : Array to store 1, 2, … , n-itemsets
//NDC: Array to store their frequencies
//CC : Calculate confidence of generated itemsets
//TransDB : Transactional Database
//Min_Supp : Minimum Support
//Min_Conf: Minimum Confidence
Input : TransDB, Min_Supp, Min_Conf
Output: Association Rules exhibiting relationships among
        Itemsets
 for each transaction Ti in database D do
     for each Ii ∈Ti do
       SISi← {Ii}      //  Add to single item set
         for k←2; k<powerset(SISi); k←k+1 do
              is ←$SIS_i$ || $SIS_i$; //Concatenate single itemsets to
              generate multiple itemsets
             if is∈NDi
                then $NDC_i$←$NDC_i$+1;//Add frequencies
                else $ND_i$ ← is; $NDC_i$ ← 1;// Count frequencies
             endif
         end// for k
     end//for each Ii
     for each itemset in NDi do
         if $NDC_i$ < Min_Support
             then Eliminate  $ND_i$ and $NDC_i$
         end
    end
     for each itemset in $ND_i$ do
           Generate antecedent and consequent
           AR ← Antecedent →=> Consequent
           CC←Min_Support(AR)/MinSupport(Antecedent)
            if CC ≥ Min_Conf
               then AR[] ←Association Rule
               else Rejection
           end
       end
end SingletonApriori
```

It is noted that in the proposed Singleton Apriori, the number of the retrieval of single itemsets on database depends on the number of transactions and the number of items in each of these transactions. It is represented as ***NS*** ∞ ***NI*** of ***NT***, where ***NS*** is Number of the retrieval of single itemsets, ***NT*** is Number of transactions and *NI* is the number of items in the transaction.

To illustrate the concept, let there are n transactions denoted as T_i and for each transaction, there are k items, j = 1, 2,…, k where k is the maximum number of items in the database.

3.1 *The Scanning Process*

Let T_i represents transaction *i*, where 1≤i≤n and I_j are itemsets with 1≤ j≤ k. Let I_{in} represents the item n in i^{th} transaction. Table 1 shows few transactions and their corresponding itemsets.

Table 1. Transaction Table

Transactions	Itemsets
T_i	I_{i3},I_{i6}
T_j	I_{j1}, I_{j5}, I_{j6}
T_k	I_{k1},I_{k4}
T_l	I_{l3}
T_m	I_{m1},I_{m2},I_{m4},I_{m5}

For illustrating the scanning process, consider the transactions T_i and T_j only. While scanning the transaction $T_{i,}$ the proposed singleton Apriori finds {I_{i3}} and {I_{i6}} as single Itemsets and then it joins single itemsets to find 2, 3,...,n-itemsets. They are {I_{i3}},{I_{i6}} and {I_{i3}, I_{i6}} taken with their frequencies are considered as F_{i3}, F_{i6} and F_{i36} respectively. The frequencies F_{i3} and F_{i6} represents the number of occurrences of the itemsets 3 and 6 respectively in the transaction i. The frequency F_{i36} is obtained by summing up the frequencies of both itemsets 3 and 6 in the transaction i. This is because to avoid the generation of separate candidate itemsets. Similarly, for scanning the transaction T_j, the single itemsets are {I_{j1}}, {I_{j5}},{I_{j6}} with frequencies F_{j1}, F_{j5}, F_{j5} respectively, the 2-itemsets are {I_{j1},I_{j5}},{I_{j1},I_{j6}},{I_{j5},I_{j6}} with frequencies F_{j15}, F_{j16}, F_{j56} respectively and the 3-itemsets are {I_{j1}, I_{j5}, I_{j6}} with frequencies F_{j156}, Table 2 and Table 3 show the itemsets generated from T_i and T_j respectively.

Table 2. Itemsets from T_i

Itemsets	Freq
I_{i3}	F_{i3}
I_{i6}	F_{i6}
I_{i3},I_{i6}	Fi_{36}

Table 3. Itemsets from T_j

Itemsets	Freq	Itemsets	Freq
I_{j1}	F_{j1}	I_{j1}, I_{j6}	F_{j16}
I_{j5}	F_{j5}	I_{j5}, I_{j6}	F_{j56}
I_{j6}	F_{j6}	I_{j1}, I_{j5}, I_{j6}	F_{j156}
I_{j1}, I_{j5}	F_{j15}		

It is noted that separate candidate itemsets are not generated during and even after the single itemsets have been generated. Thus, there is no need to store them which results in no need for memory in storing the itemsets. The scans are done only for extracting single itemsets from transactions and not for checking whether a 2-itemset, 3-itemset,…, n-itemsets are present or not. This process is repeated until all itemsets are generated from all transactions in the database. Thus, the proposed singleton apriori generates associations with a single scan to database for each transaction and also without generating candidate itemsets.

IV. Proposed Work- Example

To show the relevance of work, let n = 10 i.e., T_i, i=1,…,10 are transactions and number of itemsets k=4 i.e., $I_{j,}$ j = 1,..,4. It is shown in Table 4.

Table 4. Transaction Table

Transactions	Itemsets	Transactions	Itemsets
T_1	I_1,I_2,I_3,I_4	T_6	I_1,I_2,I_3,I_4,I_5
T_2	I_2,I_4	T_7	I_2,I_3,I_5
T_3	I_1,I_4	T_8	I_4
T_4	I_3	T_9	I_1,I_3,I_4
T_5	I_1,I_2,I_4,I_5	T_{10}	I_2,I_3,I_4,I_5

Table 5. Itemsets from T_1

Item	Freq
I_1	1
I_2	1
I_3	1
I_4	1
I_1,I_2	1
I_1,I_3	1
I_1,I_4	1
I_2,I_3	1
I_2,I_4	1
I_3,I_4	1
I_1,I_2,I_3	1
I_1,I_2,I_4	1
I_1,I_3,I_4	1
I_2,I_3,I_4	1
I_1,I_2,I_3,I_4	1

Table 6. Itemsets from T_2

Item	Freq
I_1	1
I_2	1+1
I_3	1
I_4	1+1
I_1,I_2	1
I_1,I_3	1
I_1,I_4	1
I_2,I_3	1
I_2,I_4	1+1
I_3,I_4	1
I_1,I_2,I_3	1
I_1,I_2,I_4	1
I_1,I_3,I_4	1
I_2,I_3,I_4	1
I_1,I_2,I_3,I_4	1

When the process is initiated, transaction T_1 is scanned to obtain only single items with their frequencies and 2, 3,…, n–itemsets are generated along with their frequencies from the single itemsets. They are shown in Table 5. In the subsequent processes, all other transactions are scanned and if an already existing itemset has encountered in a scanned transaction, only the frequency will be incremented and if not, the itemset is considered as new with the frequency 1. Table 6 shows the results obtained by scanning 2^{nd} transaction and Table 7 shows the results obtained by scanning all 10 transactions. It is noted that the generated itemsets are those itemsets which present in the database. As the candidate itemsets are not created in this process, memory utilization is substantially minimized. To generate frequent itemsets, the *min_support* is taken as 3. Table 8 shows the itemsets which satisfies the minimum support ≤ 3.

Table 7. Itemsets Generated from T_1 To T_{10} Transactions

Itemsets	Freq	Itemsets	Freq	Itemsets	Freq
I_1	5	I_1,I_2,I_3	2	I_1,I_2,I_5	2
I_2	6	I_1,I_2,I_4	3	I_2,I_4,I_5	3
I_3	6	I_1,I_3,I_4	3	I_1,I_2,I_4,I_5	2
I_4	8	I_2,I_3,I_4	3	I_3,I_5	3
I_1,I_2	3	I_1,I_2,I_3,I_4	2	I_1,I_3,I_5	1
I_1,I_3	3	I_5	5	I_2,I_3,I_5	3
I_1,I_4	5	I_1,I_5	2	I_1,I_2,I_3,I_5	1
I_2,I_3	4	I_2,I_5	4	I_2,I_3,I_4,I_5	2
I_2,I_4	5	I_4,I_5	4	I_1,I_2,I_3,I_4,I_5	1
I_3,I_4	4				

Table 8. Itemsets satisfying Min_Support

Itemsets	Freq	Itemsets	Freq	Itemsets	Freq
I_1	5	I_3,I_4	4	I_1,I_4	5
I_2	6	I_1,I_2,I_4	3	I_2,I_3	4
I_3	6	I_1,I_3,I_4	3	I_2,I_4	5
I_4	8	I_2,I_3,I_4	3	I_2,I_3,I_5	3
I_1,I_2	3	I_5	5	I_2,I_4,I_5	3
I_1,I_3	3	I_2,I_5	4	I_3,I_5	3
		I_4,I_5	4		

To generate the association rules, let the min_confidence ≥ 60% are taken and Table 9 shows the association rules generated based on minimum confidence.

Table 9. Associations satisfying Min_Confidence

Rules	Confidence	Rules	Confidence
$I_1 \rightarrow I_4$	100%	$I_5 \rightarrow I_4$	80%
$I_4 \rightarrow I_1$	62.5%	$I_5 \rightarrow I_2I_3$	60%
$I_2 \rightarrow I_3$	66.7%	$I_2I_3 \rightarrow I_5$	75%
$I_3 \rightarrow I_2$	66.7%	$I_3I_5 \rightarrow I_2$	100%
$I_2 \rightarrow I_4$	83.3%	$I_2I_5 \rightarrow I_3$	75%
$I_4 \rightarrow I_2$	62.5%	$I_5 \rightarrow I_2I_4$	60%
$I_2 \rightarrow I_5$	66.7%	$I_2I_4 \rightarrow I_5$	60%
$I_5 \rightarrow I_2$	80%	$I_2\ I_5 \rightarrow I_4$	75%
$I_5 \rightarrow I_3$	60%	$I_4I_5 \rightarrow I_2$	75%

V. Results and Discussion

Few challenges of traditional apriori are generation of candidate itemsets and multiple scans on the database. Multiple scans are required to generate each 1, 2, 3, ... , n candidate itemsets which leads to more processing time and requires memory space. The proposed Singleton Apriori resolves the influence of these challenges. It scans the database only once to discover 1-frequent itemsets by accumulating the count of each itemset. No further scanning is required for the generation of 2, 3,...,n frequent itemsets. Frequent itemsets from 2 to n are discovered from 1-frequent itemsets. Usage of 1-frequent itemsets for the discovery of the subsequent frequent itemsets eliminates the generation of candidate itemsets as well as multiple scans of database. As candidate itemsets are not generated, memory space is required only to store frequent itemsets.

From Table 9, it is observed that the proposed Singleton Apriori minimizes the number of scans on database and it eradicates the generation of separate 1, 2, 3,…,n- candidate itemsets to generate specific patterns. The conventional Apriori algorithm finds candidate itemsets for the generation of frequent itemsets and further scans the database for the determination of 1, 2, 3,…, n-frequent itemsets. But the proposed Singleton Apriori never generates the candidate itemsets, as an itemset has no contribution in determination of its frequency because it does not occur in a transaction. Due to this fact, it finds 1, 2, 3, ...,n-itemsets only if they occur in transactions. Besides, it scans database only once for the entire transaction to find the existence of itemsets. Frequent 1, 2, 3, ..., n itemsets are discovered from the existence of itemsets bin the first scan only. The proposed algorithm eliminates the generation of candidate itemsets and it finds frequent itemsets directly from the transactions which results in minimizing the number of scans on database i.e., the number of scans is directly proportional to the number of items of the transactions.

Fig. 1 shows the comparison of efficacy between the conventional Apriori and the proposed Singleton Apriori for a single itemset generation. Conventional Apriori scans each transaction of database for the total number of items occurs in the entire database. i.e., the number of scans is directly proportional to total number of items which occur in the entire database, whereas the proposed Singleton Apriori scans each transaction for the number of items which occurs in the same transaction. i.e., number of scans is directly proportional to total number of items occur in the corresponding transaction. This is only for single itemsets. Conventional Apriori scans each transaction twice for each 2-itemsets. Thus for n-itemsets, conventional Apriori scans each transaction n times for each n-itemset whereas the proposed algorithm scans only once for single itemsets which results in high efficacy than that of the conventional one.

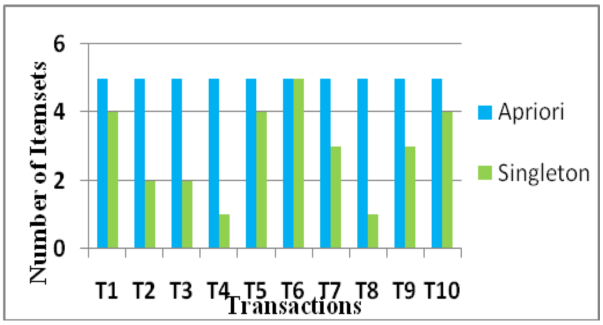

Fig.1. Efficacy Comparison in terms of time based on scans

Fig.2 shows the comparison of number of scans needed in generating the 1, 2, 3,…,n-candidate itemsets and the number of itemsets present in the transactions respectively by scanning the database based on conventional Apriori and Singleton Apriori. In conventional Apriori, the database is scanned each time to find 1, 2, 3,..., n candidate itemsets. Thus, the number of scans is determined by the number of 1, 2, 3,…, n candidate itemsets whereas in the proposed Singleton Apriori, the database is scanned only once to find 1-itemsets with the number of occurrences. 2, 3,...,n itemsets are found from 1-itemsets. Thus, the number of scans is determined by the total number of items present in the transactions. As the number of items in any transaction is not always equal to the total number of items in the entire database and also most of the times it is less than the total number of items in the database, the proposed singleton apriori always requires less number of scans on database.

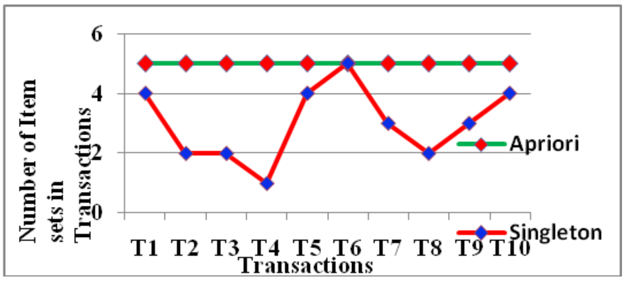

Fig.2. Comparison in terms of the number of itemsets

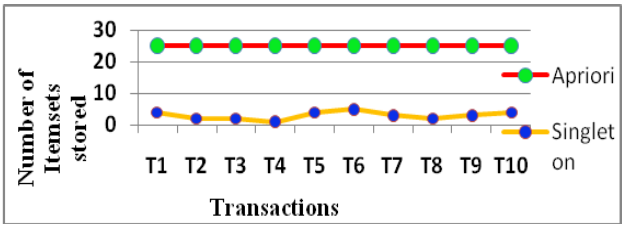

Fig.3. Efficacy on memory space

Fig. 3 represents the memory occupied by conventional Apriori and Singleton Apriori. In conventional Apriori, once the 1, 2, …, n candidate itemsets are generated, they must be preserved until all transactions are scanned to find out the frequencies of those itemsets and also 1, 2, 3, ... , n frequent itemsets, whereas Singleton apriori minimizes memory requirement because it does not generate candidate itemsets and generates only frequent itemsets.. It also requires only buffers to store the itemsets those occur in transactions and also the number of occurred itemsets never greater than the number of candidate itemsets. Thus, the proposed singleton apriori minimizes memory utilization too.

VI. CONCLUSION

A novel singleton Apriori has been proposed in this paper that scans all items in a transaction only once which results in minimizing the scanning time. Further, in the proposed work as no candidate itemsets are generated, space is not needed to store them until all transactions are scanned. The implementation results clearly show that these features enhance the overall performance both in terms of memory and speed for the generation of association rules. Further, the proposed work is completely new and innovative.

REFERENCES

[1] Jiawei Han and Michelin Kamber,"*Data Mining Concepts and Techniques*", 2nd Ed., Morgan Kaufmann Publisher, 2006.

[2] Peng Peng, Qianli Ma and Chaoxiong Li ,"The Research and Implementation of Data Mining Component Library System", College of Computer Science and Engineering, South China University of Technology, Guangzhou, PR China.

[3] Caixian Chen, Huijian Han and Zheng Liu,"KNN question classification method based on Apriori algorithm", School of Computer Science and Technology, Shandong University of Finance and Economics, Jinan, Shandong, China, March 2014.

[4] Arti Rathod, Mr. Ajaysingh Dhabariya and Chintan Thacker,"A Survey on Association Rule Mining for Market Basket Analysis and Apriori Algorithm", International Journal of Research in Advent Technology, Vol.2, March 2014.

[5] Zhi Liu, Tianhong Sun and Guoming Sang, "An Algorithm of Association Rules Mining in Large Databases Based on Sampling", International Journal of Database Theory and Application, Vol.6, 2013.

[6] Sadegh Bafandeh Imandoust and Mohammad Bolandraftar," Application of K-Nearest Neighbor (KNN) Approach for Predicting Economic Events: Theoretical Background", Journal of Engineering Research and Application,Vol. 3, Issue 5, Sep-Oct 2013.

[7] Harpreet Singh and Renu Dhir, "A New Efficient Matrix Based Frequent Itemset Mining Algorithm with Tags", International Journal of Future Computer and Communication, Vol. 2, No. 4, August 2013.

[8] Sunitha Vanamala, L.Padma sree and S.Durga Bhavani,"Efficient Rare Association Rule Mining Algorithm", International Journal of Engineering Research and Applications, Vol. 3, May-Jun 2013.

[9] Jogi.Suresh and T.Ramanjaneyulu, "Mining Frequent Itemsets Using Apriori Algorithm", International Journal of Computer Trends and Technology, Vol. 4, April 2013.

[10] P. Ajith, B. Tejaswi and M.S.S.Sai,"Rule Mining Framework for Students Performance Evaluation", International Journal of Soft Computing and Engineering, Vol. 2, January 2013.

[11] Ziauddin, Shahid Kammal, Khaiuz Zaman Khan and Muhammad Ijaz Khan, "Research on Association Rule Mining ",Advances in Computational Mathematics and its Applications ACMA, Vol. 2, 2012.

[12] Badri Patel, Vijay K Chaudhari, Rajneesh K Karan and YK Rana, "Optimization of Association Rule Mining Apriori Algorithm Using ACO", International Journal of Soft Computing and Engineering, Vol. 1, March 2011.

[13] Deepa S. Deshpande,"A Novel Approach for Association Rule Mining using Pattern Generation", International Journal of Information Technology and Computer Science, Vol. 6, No. 11, pp.59-65, October 2014, doi: 10.5815/ijitcs.2014.11.09.

[14] Darshan M. Tank, "Improved Apriori Algorithm for Mining Association Rules", International Journal of Information Technology and Computer Science, Vol. 6, pp.15-23, No. 7,June 2014, doi:10.5815/ijitcs.2014.07.03.

[15] Padam Gulwani, "Association Rule Hiding by Positions Swapping of Support and Confidence", International Journal of Information Technology and Computer Science, Vol. 4, No. 4, pp.54-61, April 2012.

[16] Sonia Setia and Dr. Jyoti, "Multi-Level Association Rule Mining: A Review ", International Journal of Computer Trends and Technology, Vol. 6, Dec 2013.

[17] Thabet Slimani and Amor Lazzez,"Efficient Analysis of Pattern and Association Rule Mining Approaches", International Journal of Information Technology and Computer Science, Vol. 03, pp. 70-81, 2014, doi:10.5815/ijitcs.2014.03.09.

[18] Manish Saggar, Ashish Kumar and Agrawal Abhimanyu Lad, Optimization of Association Rule Mining using Improved Genetic Algorithms, IEEE International Conference on Systems, Man and Cybernetics, 2004.

Author's Profiles

K. Mani received his MCA and M.Tech from the Bharathidasan University, Trichy, India in Computer Applications and Advanced Information Technology respectively. After did his MCA, he got his Graduation in Operations Research from Operational Research Society of India, Kolkatta. Since 1989, he has been with the Department of Computer Science at the Nehru Memorial College, affiliated to Bharathidasan University where he is currently working as an Associate Professor. He completed his PhD in Cryptography with primary emphasis on evolution of framework for enhancing the security and optimizing the run time in cryptographic algorithms. His current research area includes cryptography, data mining and coding theory. He published and presented around 20 research papers at international journals and conferences.

R.Akila received her B.Sc and M.Sc degrees in Computer Science from Seethalakshmi Ramaswami College, affiliated to Bharathidasan University, Tiruchirappalli, Tamilnadu, India. She received M.Phil degree in Computer Science from Mother Teresa Women's University, Kodaikanal, Tamilnadu, India. She has worked in E.M.G.Yadava Women's College, Madurai and in Cauvery College for Women, Trichy. She is presently working as an Assistant Professor in the P.G. and Research Department of Computer Science, Nehru Memorial College, Puthanampatti, Tiruchirappalli. She is pursuing PhD degree in Computer Science at Bharathidasan University. Her research interests include Data Mining techniques, Algorithms, Big Data and fuzzy systems.

How to cite this paper: K.Mani, R.Akila,"Enhancing the Performance in Generating Association Rules using Singleton Apriori", International Journal of Information Technology and Computer Science(IJITCS), Vol.9, No.1, pp.58-64, 2017. DOI: 10.5815/ijitcs.2017.01.07