



## Comparative Study of Clustering and Association Method for Large Database in Time Domain

**Mrs. Sonali Manoj Raut**

Sipna's college of Engg. And Tech  
Amravati, Maharashtra, India

**Prof. Dhananjay Dakhane**

Sipna's college of Engg. and Tech.  
Amravati, Maharashtra, India

*Abstract -- Data mining technique is now used in a wide variety of fields. Because of rapid growth in world wide information, efficiency of association rule mining has been concerned for several years. Association rules mining (ARM) algorithms have been extensively researched in the last decade. Therefore, numerous algorithms were proposed to discover frequent itemsets and then mine Association rules. Association rules used widely in the area of market basket analysis can be applied to the analysis of expression data as well. Association rules mining algorithms have been extensively researched in the last decade. Therefore, numerous algorithms were proposed to discover frequent item sets and then mine association rules. This paper will present an efficient association rules algorithm by using concept of clustering and graph based structure to find the frequent itemset. But for large database constructing the graph is not feasible so in proposed method instate of constructing the graph for finding the frequent itemset use of transitive relation. And comparing this new approach with previous algorithm for large database in time domain. The proposed method is to create a cluster table by scanning the database only once, and then clustering the transaction into clusters according to their length. The method reduces main memory requirement since it consider only a small cluster at a time and hence it is scalable and efficient for any large size of the database.*

**Keywords-** Data mining, Association rule, frequent itemset, Cluster

### I. INTRODUCTION

Due to growth of the data volume in the last decade, a set of different techniques for deletion of repetitive data and conversion of data to more usable forms has been proposed under the name of Data Mining. There exist a set of different techniques concerning the Data Mining, Some of which are decision trees, associative rules and Data Clustering. Clustering is the task of assigning a set of objects into groups so that the objects in the same cluster are more similar each other than to these in other cluster. Data mining is a tool that supports research and allows new assertions to be made by disclosing previously undisclosed details in large amounts of data [1]. One of the most challenges in database mining is developing fast and efficient algorithms that can deal with large volume of data because most mining algorithms perform computation over the entire database and mostly the databases are very large. With the ever increasing trend towards data volume growth, hardware speedup and the storage Capacity of the computers, the data-mining field researches have been more than before, tempted to capture the underlying rule, knowledge, and relations hidden in the data. There exist a set of different techniques concerning the Data Mining, the most paramount of which is Data Clustering. Association rules mining is one of the most well studied data mining tasks. It discovers relationships among attributes in different types of databases, producing if-then statements concerning attribute-values [2]. It was firstly introduced in [3] to discover association rules between items over basket data, an association rule describes the associations among items in which when some items are purchased in a transaction, the others are purchased, too. In order to find association rules, we need to discover all large or frequent itemsets from a large database of customer transactions. A large itemset is a set of items which appear often enough within the same transactions. In this paper, we introduce a technique called Clustering and graph based association.

The discovery of association relationship among huge database has been known to be useful in selective marketing, decision analysis and business management [4]. A popular area of application is the market basket analysis, which studies the buying behaviors of customers by searching for sets of items that are frequently purchased together.

### II. ASSOCIATION RULE MINING

Association rule mining, one of the most important and well researched techniques of data mining. It aims to extract interesting correlations, frequent patterns, associations or casual structures among sets of items in the transaction databases or other data repositories. Association rules are widely used in various areas such as telecommunication networks, market and risk management, inventory control etc.

Association rule mining is to find out association rules that satisfy the predefined minimum support and confidence from a given database. The problem is usually decomposed into two sub problems. One is to find those item sets whose occurrences exceed a predefined threshold in the database; those item sets are called frequent or large item sets. Then other rule are generated by deleting the last items in the antecedent and inserting it to the consequent, further the confidences of the new rules are checked to determine the interestingness of them.

The first sub-problem can be further divided into two sub-problems: candidate large item sets generation process and frequent item sets generation process. We call those item sets whose support exceed the support threshold as large or frequent item- sets, those item sets that are expected or have the hope to be large or frequent are called candidate item sets.

In many cases, the algorithms generate an extremely large number of association rules, often in thousands or even millions. Further, the association rules are sometimes very large. It is nearly impossible for the end users to comprehend or validate such large number of complex association rules, thereby limiting the usefulness of the data mining results. Several strategies have been proposed to reduce the number of association rules, such as generating only "interesting" rules, generating only "non redundant" rules, or generating only those rules satisfying certain other criteria such as coverage, leverage, lift or strength.

An association rule is an *implication* or *if-then-rule* which is supported by data. The motivation given in for the development of association rules is *market basket analysis* which deals with the contents of point-of sale transactions of large retailers[5]. A typical association rule resulting from such a study could be "90 percent of all customers who buy bread and butter also buy milk". Insights into customer behavior may also be obtained through customer surveys, but the analysis of the transactional data has the advantage of being much cheaper and covering all current customers. Compared to customer surveys, the analysis of transactional data does have some severe limitations, however. For example, point-of-sale data typically does not contain any information about personal interests, age and occupation of customers. Nonetheless, market basket analysis can provide new insights into customer behavior and has led to higher profits through better customer relations, customer retention, better product placements, product development and fraud detection.

### III. CONCEPT OF ASSOCIATION MINING

**Item :** It is a field of the transaction database.

**Transaction :**It is corresponding to a record of the database. Transaction usually is marked as small letter t to mark item i.  $t_i = \{i_1, i_2, \dots, i_p\}$ . Each transaction has an only identifier called TID. The whole set of transaction  $t_i$  constitutes a database D.  $D = \{t_1, t_2, \dots, t_n\}$

**Support:**The support of association rule  $X \rightarrow Y$  in transaction database is a ratio. The ratio is between the count of item set which contains X and Y, and the count of all of item set. That marks  $\text{support}(X \rightarrow Y)$ . That is the percent of the item set containing X and Y at the same time in the transaction database.

**Confidence:** It is the ratio between the count of transaction containing X and Y and the count of transaction containing X. That is marked as  $\text{confidence}(X \rightarrow Y)$ . Confidence is the percent of the transaction sets containing X and Y at the same time in the transaction database.

**Frequent Item set:** The item set, whose support is not lower than the minimum support (Min Sup).

**Strong rule and Weak rule:** If  $\text{support}(X \rightarrow Y) \geq \text{MinSupport}$  and  $\text{Confidence}(X \rightarrow Y) \geq \text{MinConf}$ , then mark association rule  $X \rightarrow Y$  as strong rule, otherwise mark it as a weak rule.

### IV. SEARCHING FREQUENT ITEMSET

Frequent patterns, such as frequent item sets, substructures, sequences term-sets, phrase-sets, and sub graphs, generally exist in real-world databases. Identifying frequent item sets is one of the most important issues faced by the knowledge discovery and data mining community. Frequent item set mining plays an important role in several data mining fields as association rules [6] warehousing, correlations, clustering of high-dimensional biological data, and classification. Given a data set d that contains k items, the number of item sets that could be generated is  $2^k - 1$ , excluding the empty set[6]. In order to searching the frequent item sets, the support of each item sets must be computed by scanning each transaction in the dataset. A brute force approach for doing this will be computationally expensive due to the exponential number of item sets whose support counts must be determined. There have been a lot of excellent algorithms developed for extracting frequent item sets in very large databases. The efficiency of algorithm is linked to the size of the database which is amenable to be treated. There are two typical strategies adopted by these algorithms: the first is an effective pruning strategy to reduce the combinational search space of candidate item sets (Apriori techniques). The second strategy is to use a compressed data representation to facilitate in-core processing of the item sets (FP-tree techniques).

### V. APRIORI ALGORITHM

In 1994 Agrawal etc. put forward famous Apriori algorithm according to the property of association rule: the sub sets of the frequent item set is also frequent item set, the supersets of non-frequent item set is also non-frequent item set. The algorithm each time makes use of k-frequent item set carrying on conjunction to get k+1 candidate item set.

The key idea of Apriori algorithm is to make multiple passes over the database. It employs an iterative approach known as a breadth-first search (level-wise search) through the search space, where  $k$ -item sets are used to explore  $(k+1)$ -item sets.

The working of Apriori algorithm is fairly depends upon the Apriori property which states that "All nonempty subsets of a frequent item sets must be frequent" [5]. It also described the anti monotonic property which says if the system cannot pass the minimum support test, all its supersets will fail to pass the test. Therefore if the one set is infrequent then all its supersets are also frequent and vice versa. This property is used to prune the infrequent candidate elements. In the beginning, the set of frequent 1-itemsets is found. The set of that contains one item, which satisfy the support threshold, is denoted by  $L_1$ . In each subsequent pass, we begin with a seed set of item sets found to be large in the previous pass. This seed set is used for generating new potentially large item sets, called candidate item sets, and count the actual support for these candidate item sets during the pass over the data. At the end of the pass, we determine which of the candidate item sets are actually large (frequent), and they become the seed for the next pass. Therefore,  $L_{k-1}$  is used to find  $C_k$ , the set of frequent 2-itemsets, which is used to find  $L_2$ , and so on, until no more frequent  $k$ -item sets can be found. The feature first invented by in Apriori algorithm is used by the many algorithms for frequent pattern generation. The basic steps to mine the frequent elements are as follows :

**1.Generate and test:** In this first find the 1-itemset frequent elements by scanning the database and removing all those elements from which cannot satisfy the minimum support criteria.

**2.Join step:** To attain the next level elements join the previous frequent elements by self join i.e. known as Cartesian product of  $L_{k-1}$ . i.e. This step generates new candidate  $k$ -item sets based on joining with itself which is found in the previous iteration. Let  $C_k$  denote candidate  $k$ -item set and  $L_{k-1}$  be the frequent  $k$ -item set.

**3.Prune step:** is the superset of so members of may or may not be frequent but all frequent item sets are included in thus prunes the  $C_k$  to find frequent item sets with the help of Apriori property. i.e. This step eliminates some of the candidate  $k$ -item sets using the Apriori property A scan of the database.

To illustrate this, suppose  $n$  frequent 1-itemsets and minimum support is 1 then according to Apriori will generate  $C_2$  and so on. The total number of candidates generated is greater than Therefore suppose there are 1000 elements then 1499500 candidate are produced in 2 itemset frequent and 166167000 are produced in 3-itemset frequent . It is no doubt that Apriori algorithm successfully finds the frequent elements from the database. But as the dimensionality of the database increase with the number of items then:

1) More search space is needed and I/O cost will increase.

2) Number of database scan is increased thus candidate generation will increase results in increase in computational cost.

Therefore many variations have been takes place in the Apriori algorithm to minimize the above limitations arises due to increase in size of database. These subsequently proposed algorithms adopt similar database scan level by level as in Apriori algorithm, while the methods of candidate generation and pruning, support counting and candidate representation may differ. The algorithms improve the Apriori algorithms by:

1) Reduce passes of transaction database scans

2) Shrink number of candidates

3) Facilitate support counting of candidates

## VI. CLUSTERING AND GRAPH-BASED ASSOCIATION RULE [5]

Clustering and graph-based association rule (CGAR), for efficient association rules mining, which overcome the drawbacks of the Apriori algorithms. CGAR which is fundamentally different from all the previous algorithms in the following points:

i. It reads the database of transaction only once to generate frequent 1-itemsets.

ii. It is scalable with all types of databases regardless to their size.

iii. It is as efficient as it requires less memory and CPU time to generate strong rules from the Transaction database.

iv. It is easy to implement as it uses simple cluster table and a robust graph data structure.

The items should be given sequential numbers to simplify the process of building the graph; this must be taken in consideration as an important action before applying proposed algorithm. CGAR scans the database of transactions only once to build the clustering table as a two-dimensional array where the columns represent items and the rows represent transactions' IDs (TIDs). The contents of the table consist of 0 or 1 to indicate the absence or presence of an item in a transaction, respectively. After that, the bit vectors for each item will be ready and it is an easy process to determine the frequent  $l$ -itemsets by counting the number of 1s in each transaction, if it isn't less than the minimum support threshold, it is considered as a frequent item set and then be used in building the graph, otherwise, it will be discarded from further discussion as it is infrequent item. The second phase starts by reordering frequent 1-itemsets by providing each one with a sequential number to facilitate the process of constructing the graph, which is constructed by doing logical and operation between each pair of consecutive frequent 1-

Item sets  $\langle item_i, item_j \rangle \mid i < j$ , if the number of 1s in the result is greater than or equal to minimum support threshold, a directed edge is drawn from  $item_i$  to  $item_j$ , this operation is repeated for all frequent 1-itemsets. As the graph is completed,

the set of frequent 2-itemsets are generated, and it will be direct from the graph traversing to generate frequent k-itemsets, such as  $k \geq 3$ .

In the standard situation, as the database contains hundreds of thousands of transactions and different items, constructing only one graph is not practical, and so it has been suggested to construct different graphs for each cluster and find from this graph all frequent item sets, then combine the subsets of frequent item sets together to get the whole set of frequent item sets, and this technique is scalable with all transactions databases of different sizes.

When the database contains hundreds of thousands of transactions and different items, constructing only one graph is not practical, hence there is requirement of different graphs for each cluster and finding from this graph all frequent item sets. Then combine the subsets of frequent item sets together to get the whole set of frequent item sets which is time consuming process.

## VII. PROPOSED SYSTEM

Overcome the problem of graph construction, an improvement in CGAR is suggested.

### Reduction in Step of finding 1 frequent itemset

Instead of counting all 1's in bit vector just count number of 1's. While counting number of 1's if the number of 1's exceed minimum support stop further counting 1's. This will reduce time in generation of 1 frequent itemsets.

### Reduction in step of finding 2 frequent itemset

When we are finding 2 frequent itemset if any of the items support is 100%, there is no need of combing that item with other items and finding the support. simply combine item with 100% support with others. This step will reduce time for generation of 2 frequent itemsets.

### Finding 3 Frequent Item set Generation

Instate of using the graph for finding frequent 3 itemset use of transitive relation. So it is fisible for large size of database. Because constructing the single graph for large database is not fisible.

## VIII. EXPERIMENTAL RESULT

The existing and proposed methods are implemented in vb 6.0. The experiments were performed on a 1.8GHz Intel Core 2 Duo CPU PC with 2GB main memory, running on Microsoft Windows XP professional operating system. Following are the graphs drawn based on results observed. Data set of different size (no. of records) is used for testing performance of Apriori algorithm and proposed method.

### Apriori vs. proposed method with fixed support and variable Datasets

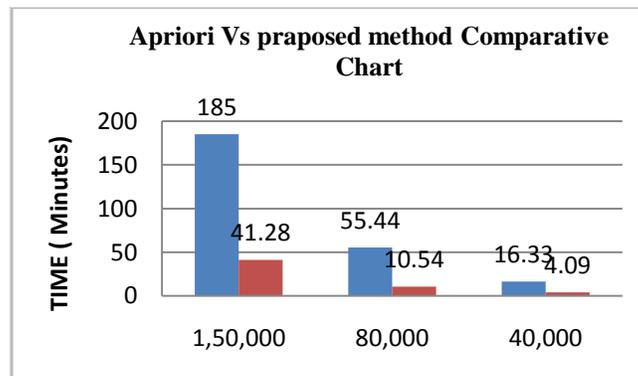
Here overall performance of both the algorithms is compared. Overall performance includes generation of association rules in apriori and proposed method based on fixed support value given by user. In below graph support is 30%.

#### Graph :

Data Size Range : 40000 to 150000 no. of lines in input file.

Time varies from 1 to 190 minutes.

Support : 30 %



#### Observations & analysis:

Following observations are based on graphs

1. As the dataset size goes from 10k to 15k, the runtime of overall performance of proposed method is less than Apriori algorithm. There is a slight difference in time required by both the methods when the dataset is 2k.
2. As the dataset size goes from 20k to 15k, the runtime of overall performance of Apriori and proposed method can be seen with great difference.
3. For dataset of larger size i.e 15k proposed method performs very well as compared to Apriori method.

## IX. CONCLUSION

The existing and proposed methods are implemented in vb 6.0. The experiments were performed on a 1.8GHz Intel Core 2 Duo CPU PC with 2GB main memory, running on Microsoft Windows XP professional operating system.. Data set of different size (no. of records) is used for testing performance of Apriori algorithm and proposed method algorithm.

In this paper the traditional data mining method such as Apriori is discussed. Although these techniques give the efficient performance, they cannot handle the real world dynamic data. As data size increases efficiency of Apriori decreases as it needs a complete database scan each time. So proposed method technique is introduced which solves the problem of rescanning. Experimental results proposed method has outperformed the apriori method Based on fact that number of scans requirement of Apriori and CGAR method are multiple and one respectively, it is proved that proposed method can outperform Apriori method .

## **REFERENCES**

- [1] Ben franklin, Genealogical data Mining,2006 .
- [2] R. Agrawal, T. Imilienski, A. Swami, Mining association rules between sets of items in large databases, Proceedings of the ACM SIGMOD International Conference on Management of Data, Washington, DC, 1993 pp. 207–216.
- [3] Yuh-Jiuan Tsay, Jiunn-Yann Chiang, CBAR: an efficient method for mining association rules, Knowledge-Based Systems 18 (2005) 99–105.
- [4] M.S. Chen, J.Han and P.S.Yu, "Data mining: an overview from Database Perspective", IEEE trans, Knowledge and Data eng., vol.8,no.6,pp.866-883,Dec.1996}, [16] J.Hipp, U.Guntzer, and G.Nakhaeizadeh, "algorithm for association Rule Mining –A general survey and comparision," ACM, SIGKDD explorations vol.2,no.1 ,pp,58-64 July 2000
- [5] Wael A. AlZoubi, Azuraliza Abu Bakar, Khairuddin Omar, " Scalable and Efficient Method for Mining Association Rules ", Proc. Of IEEE, vol. 1,no.8, 2009, pp. 36-41.
- [6] Lijuan Zhou, Shuang Li, Mingsheng Xu "Research on Algorithm of Association Rules in Distributed Database System", Proc. Of IEEE, vol. 3,no.9, 2010, pp. 216-219