International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347 - 6435 (Online) Volume 2, Issue 5, May 2014)

Review of Association Rule in Data Mining Algorithm

Kavita[1], Trilok Gabba[2]
[1]*M. Tech Scholar,* [2]*Head of Department, BITS, Bhiwani*

***Abstract*— now data in organizations is increasing very speedily. A technique is required for handling this data. So there are many techniques available in market that organizes this data. Some operations are applied on data like classification, clustering and association rule. One of the most popular algorithms is Apriori that is used to extract frequent item sets from large database and getting the association rule for discovering the knowledge. This paper represents review of algorithms that are used for association rule mining.**

***Keywords*— Data Mining, Association Rule, Apriori Algorithm, Item set generation, Railway traffic.**

I. INTRODUCTION

Many business enterprises accumulate large quantities of data from their daily operations. As in example, customer purchase data in huge amount are collected daily at the checkout counters of grocery stores. Table 1 describes such data example, commonly known as market transactions of basket. Transaction represents by each row in given table, which contains a unique identifier labeled T ID and a set of items bought by a customer. Data can be analyzed if retailers are interested to learn about the purchasing behavior of their customers. This important and valuable information can be used to support a variety of business-related applications such as c-r relationship, inventory management, and customer relationship management.

Table -1
An Example Of Market Basket Transactions

T ID	Item
1	{Bread, Milk}
2	{Bread, Diapers, Beer, Eggs}
3	{Milk, Diapers, Beer, Cola}
4	{Bread, Milk, Diapers, Beer}
5	{Bread, Milk, Diapers, Cola}

For example, the following rule can be extracted from the data set shown in Table 1:

{Diapers} −→ {Beer}

The rule suggests that a strong relationship exists between the sale of diapers and beer because many customers who buy diapers may be buy beer.

These types of rules use by retailers to help them identify new opportunities for cross selling their products to the customers. Besides market basket data, association analysis is also applicable to other application domains such as Web mining, and scientific data analysis. In the analysis of Earth science data, for example, the association patterns may reveal interesting connections around ocean, land, and process of atmosphere. Such information may help Earth scientists develop a better understanding of how the different elements of the Earth system interact with each other. Generally applicable to a wider variety of data sets technique is presented here, for illustrative purposes, our discussion will focus mainly on market basket data. There are two key issues that need to be addressed when applying association analysis to data of basket. First, a patter can be discovering from a large transaction data set that is computationally expensive. Second, as we discover pattern some of them are potentially best because they may happen simply by chance.

II. FREQUENT ITEM-SET GENERATION

A lattice structure can be used to enumerate the list of all possible item sets. Figure 6.1 shows an item set lattice for I= {a, b, c, d, e}. In general, a data set that contains k items can potentially generate up to 2k − 1 frequent item sets, excluding the null set [1]. Because k can be very large in many practical applications, the search space of item sets that need to be explored is exponentially large

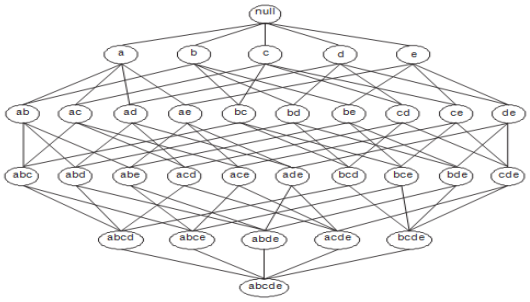

Figure1. An item-set lattice

A brute-force approach for finding frequent item sets is to determine the support count for every candidate item set in the lattice structure [2].

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347 - 6435 (Online) Volume 2, Issue 5, May 2014)

To do this, we need to compare each candidate against every transaction, an operation that is shown in Figure 2. If the candidate is contained in a transaction, its support count will be incremented. For example, the support for {Bread, Milk} is incremented three times because the item set is contained in transactions 1, 4, and 5. Such an approach can be very expensive because it requires O (N M w) comparisons, where N is the number of transactions, M = 2k − 1 is the number of candidate item sets, and w is the maximum transaction width.

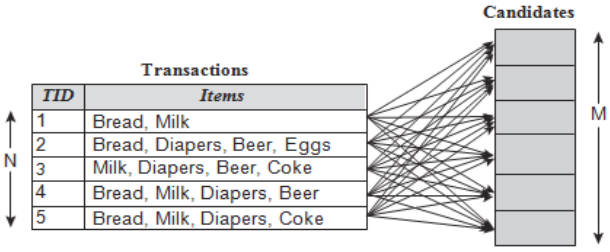

Figure2. Counting the support of candidate item-sets

Computational complexity can be reduced by several ways [3] of frequent item set generation.

1. Reduce the number of candidate item-sets (M). The Apriori principle, described in the next section, is an effective way to eliminate some of the candidate item sets without counting their support values.
2. Reduce the number of comparisons. Instead of matching each candidate item set against every transaction, we can reduce the number of comparisons by using more advanced data structures, either to store the candidate item sets or to compress the data set [4].

III. The Apriori Principle

This section describes how the support measure helps to reduce the number of candidate item-sets explored during frequent item-set generation. The use of support for pruning candidate item-sets is guided by the following principle. Theorem 1 (Apriori Principle) [5]. If an item-set is frequent, then all of its subsets must also be frequent.

To illustrate the idea behind the Apriori principle, consider the item-set lattice shown in Figure 3. Suppose {c, d, e} is a frequent item-set. Clearly, any transaction that contains {c, d, e} must also contain its subsets, {c, d}, {c, e}, {d, e}, {c}, {d}, and {e}. As a result, if {c, d, e} is frequent, then all subsets of {c, d, e} (i.e., the shaded item-sets in this figure) must also be frequent.

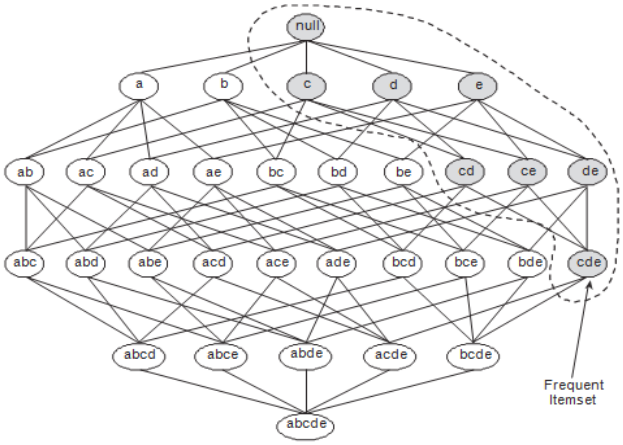

Figure3. an illustration of the Apriori principle. If {c, d, e} is frequent, then all subsets of this Item-set are frequent.

Conversely, if an item-set such as {a, b} is infrequent, then all of its supersets must be infrequent too. As illustrated in Figure 4, the entire sub-graph containing the supersets of {a, b} can be pruned immediately once {a, b} is found to be infrequent. This strategy of trimming the exponential search space based on the support measure is known as support-based pruning.

Such a pruning strategy is made possible by a key property of the support measure, namely, that the support for an item-set never exceeds the support for its subsets. This property is also known as the anti-monotone property of the support measure.

Definition 1 (Monotonicity Property): Let I be a set of items, and J = 2I be the power set of I. A measure f is monotone (or upward closed) if

$$\forall X, Y \in J: (X \subseteq Y) \rightarrow f(X) \leq f(Y),$$

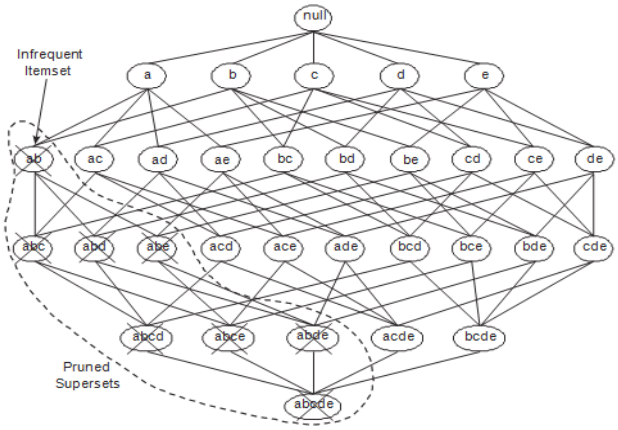

Figure4. An illustration of support-based pruning. If {a, b} is infrequent, then all supersets of {a, b} are infrequent.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347 - 6435 (Online) Volume 2, Issue 5, May 2014)

Which means that if X is a subset of Y , then f (X) must not exceed f (Y). On the other hand, f is anti-monotone (or downward closed) if

$$\forall X, Y \in J: (X \subseteq Y) \longrightarrow f(Y) \leq f(X),$$

This means that if X is a subset of Y, then f(Y) must not exceed f(X).

IV. Pseudocode For The Frequent Itemset Generation

The pseudo code for the frequent item-set generation part of the Apriori algorithm [6]. Let Ck denote the set of candidate k-item-sets and Fk-denote the set of frequent k-item-sets:

- The algorithm initially makes a single pass over the data set to determine the support of each item. Upon completion of this step, the set of all frequent 1-itemsets, F1, will be known (steps 1 and 2).
- Next, the algorithm will iteratively generate new candidate k-item-sets using the frequent (k − 1)-item-sets found in the previous iteration (step5). Candidate generation is implemented using a function called apriorigen

Algorithm 1 Frequent itemset generation of the Apriori algorithm.

1: k = 1.

2: Fk= { i | i ∈ I ∧ σ({i}) ≥ N × minsup}. {Find all frequent 1-itemsets}

3: repeat

4: k = k + 1.

5: Ck = apriori-gen (Fk−1). {Generate candidate itemsets}

6: for each transaction t ∈ T do

7: Ct = subset (Ck, t). {Identify all candidates that belong to t}

8: for each candidate itemset c ∈ Ct do

9: σ(c) = σ(c) + 1. {Increment support count}

10: end for

11: end for

12: Fk= { c | c ∈ Ck ∧ σ(c) ≥ N × minsup}. {Extract the frequent k-itemsets}

13: until Fk= ∅

14: Result = Fk.

- To count the support of the candidates, an additional pass over the data set is required for this algorithm (steps 6–10). All the candidate item-sets in Ck are determined by subset function that is contained in each transaction t [7].
- After that the algorithm eliminates all candidate item-sets whose support counts are less than minsup (step 12).
- The algorithm terminates when there are no new frequent item-sets generated, i.e., Fk= ∅ (step 13).

The frequent item-set generation part of the Apriori algorithm has two important characteristics. First, it traverses the item-set lattice one level at a time, from frequent 1-itemsets to the maximum size of frequent item-sets so it is level wise. Second, it employs a generate-and-test strategy for finding frequent item-sets. At each iteration, new candidate item-sets are generated from the frequent item-sets found in the previous iteration. Support counted and tested against the minsup threshold for each candidate. The total number of iterations needed by the algorithm is kmax + 1, where kmaxis the maximum size of the frequent item-sets.

V. Conclusion

In this paper, we have studied about association rule in data mining. Our main focus on apriori algorithm that is technique of association rule. Implementation and code used for implementation is described in this paper in section II and III. Finally with study we have conclude that there is problem of FSC (Frequent Set Counting) is identified that can be solve by enhancement in apriori algorithm. This review paper is base for research in apriori algorithm in future.

REFERENCES

[1] R. C. Agarwal, C. C. Aggarwal, and V. V. V. Prasad. “A Tree Projection Algorithm for Generation of Frequent Itemsets”. Journal of Parallel and Distributed Computing (Special Issue on High Performance Data Mining), 61(3):350–371, 2001.

[2] R. C. Agarwal and J. C. Shafer. “Parallel Mining of Association Rules”. IEEE Transactions on Knowledge and Data Engineering, 8(6):962–969, March 1998..

[3] C. C. Aggarwal and P. S. Yu. “Mining Large Itemsets for Association Rules”. Data Engineering Bulletin, 21(1):23–31, March 1998.

[4] C. C. Aggarwal and P. S. Yu. "Mining Associations with the Collective Strength Approach". IEEE Trans. on Knowledge and Data Engineering, 13(6):863–873, January/February 2001..

[5] R. Bayardo and R. Agrawal. Mining the Most Interesting Rules. In Proc. of the 5th Intl. Conf. on Knowledge Discovery and Data Mining, pages 145–153, San Diego, CA, August 1999.

[6] E.-H. Han, G. Karypis, and V. Kumar. "Min-Apriori: An Algorithm for Finding Association Rules in Data with Continuous Attributes". http://www.cs.umn.edu/˜han, 1997.

[7] G. Dong and J. Li. Efficient Mining of Emerging Patterns: Discovering Trends and Differences. In Proc. of the 5th Intl. Conf. on Knowledge Discovery and Data Mining, pages 43–52, San Diego, CA, August 1999.