Journal of Computer Science, 9 (1): 55-62, 2013
ISSN 1549-3636

doi:10.3844/jcssp.2013.55.62 Published Online 9 (1) 2013 (http://www.thescipub.com/jcs.toc)

An Efficient Prediction of Missing Itemset in Shopping Cart

[1]Nirmala, M. and [2]V. Palanisamy

[1]Department of Computer Science and Engineering,
[2]Department of Electronics and Communication Engineering,
Info Institute of Engineering, Coimbatore, Tamilnadu, India

Received 2012-09-15; Revised 2012-10-03; Accepted 2013-02-08

ABSTRACT

Many researches has focused mainly on how to expedite the search for frequently co-occurring groups of items in "shopping cart" and less attention has been paid to the methods that exploit these "frequent itemsets" for prediction purposes. This study contributes to this task by proposing a technique that uses the partial information about the contents of a shopping cart for the prediction of what else the customer is likely to buy. Several algorithms have been introduced to detect the frequently co occurring group of items in the transactional databases for prediction purposes. This study presents a new technique whose principal diagonal elements represent the association among items and looking to the principal diagonal elements, the customer can select what else the other items can be purchased with the current contents of the shopping cart and also reduces the rule mining cost. The association among items is shown through Graph. The frequent itemsets are generated from the Association Matrix. Then association rules are to be generated from the already generated frequent itemsets. We conducted extensive experiments and showed that the accuracy of our algorithm is higher than the previous algorithm. Our experiments show that the time needed for predicting the items is highly reduced than other algorithms. Moreover the memory requirement is also less since our work does not generate candidate itemsets. In this study, we have successfully implemented the Rule generation technique and predicted the set of other items that the customer is likely to buy. The performance of our algorithm outperforms the existing algorithm that needs multiple passes over the database in such a way that it efficiently mines the association among the items in the shopping cart and the prediction time of the items is greatly reduced.

Keywords: Association Rule Mining, Frequent Itemset, Data Mining, Prediction

1. INTRODUCTION

Data mining is the process of extracting potentially useful information from a data set. In Data Mining the task of finding frequent pattern in large databases is very important and has been studied in large scale in the past few years. Unfortunately, this task is computationally expensive, especially when a large number of patterns exist. This large number of patterns which are mined during the various approaches makes the user very difficult to identify the patterns which are very interesting for him. Data mining can be regarded as an algorithmic process that takes data as input and yields patterns, such as classification rules, itemsets, association rules, or summaries, as output. An itemset is a set (i.e., a group) of items. The goal of frequent itemset mining is to identify all frequent itemsets, i.e., itemsets that have at least a specified minimum support, the percentage of transactions containing the itemset. The rationale behind using support is that only itemsets with high frequency are of interest to users (Raghavan and Hafez, 2000). Frequent itemset mining plays an essential role in the theory and practice of

Corresponding Author: Nirmala, M., Department of Computer Science and Engineering, Info Institute of Engineering, Coimbatore, Tamilnadu, India

many important data mining tasks, such as mining association rules, long patterns, emerging patterns and dependency rules. It has been applied in fields such as telecommunications, census analysis and text analysis.

Association mining that discovers dependencies among values of an attribute was introduced and has emerged as a prominent research area (Agrawal and Srikant, 1994).There are two important basic measures for association rules, support(s) and confidence(c). Since the database is large and users concern about only those frequently purchased items, usually thresholds of support and confidence are predefined by users to drop those rules that are not so interesting or useful. The two thresholds are called minimal support and minimal confidence respectively. Support(s) of an association rule is defined as the percentage/fraction of records that contain X andY to the total number of records in the database. Suppose the support of an item is 0.1%, it means only 0.1% of the transaction contain purchasing of this item. Confidence of an association rule is defined as the percentage/fraction of the number of transactions that contain X andY to the total number of records that contain X. Several measures have been introduced to define the strength of the relationship between itemsets X and Y such as support, confidence and interest. The definitions of these measures, from a probabilistic model are given below:

- Support (X⇒Y)⇒P(X,Y), or the percentage of transactions in the database that contain both X and Y
- Confidence (X⇒Y)⇒P(X,Y)/P(X), or the percentage of transactions containing Y in transactions those contain X
- Interest(X ⇒ Y) ⇒ P(X , Y)/P(X)P(Y) represents a test of statistical independence

The frequent pattern mining algorithms use the Apriori principle (Aggarwal *et al*., 2002) that any superset of a non frequent itemset is also non-frequent. This antimonotone property of itemsets is used to reduce the search space by pruning the non-frequent itemsets early. The Apriori principle does not hold for itemset utility, since the superset of a low utility itemset may not be a low utility itemset. For example, if itemset {printer ink} has a low utility, its superset, {printer ink, color laser printer} might have a high utility and so we cannot prune {printer ink}.

The great practical benefits of mining association rules and its wide area of applications (Li *et al*., 2001; Liu *et al*., 2002) have led to several proposals for fast mining of association rules. The discovery of such associations can help retailers develop marketing strategies by gaining insight into which items are frequently purchased together by customer and which items bring them better profits when placed with in close proximity.

The two types of finding association between products existing in a large database are Boolean (Bollmann-Sdorra *et al*., 2001) and Quantitative. Boolean association rule mining finds association for the entire dataset. Quantitative association rule mining finds association for the clusters formed from the dataset.

In recent years, association rule mining is studied deeply in data mining field for its widely application, including financial analysis, the retail industry and business decision-making. Traditionally, an association rule is interesting if its support and confidence values are not less than thresholds given.

1.1. Related Works

Association rule mining systems that have been developed with classification purposes (Aggarwal *et al*., 2002) in mind are sometimes dubbed classification rule mining. Some of these techniques can be adapted to our needs.

For instance, one existing approach says that if i_j is the item whose absence or presence is to be predicted, the technique can be used to generate all rules that have the form:

$$\bar{r}^{(a)} \Rightarrow I_j, \text{where } \bar{r}^{(a)} \subseteq \left(I \setminus \{i_j\}\right)$$

and I_j is the binary class label (i_j = present or i_j = absent). For a given itemset s, the technique identifies among the rules with antecedents subsumed by s those that have the highest precedence according to the reliability of the rules-this reliability is assessed based on the rules' confidence and support values. The rule is then used for the prediction of i_j. The method suffers from three shortcomings. First, it is clearly not suitable in domains with many distinct items i_j. Second, the consequent is predicted based on the "testimony" of a single rule, ignoring the simple fact that rules with the same antecedent can imply different consequents-a method to combine these rules is needed. Third, the system may be sensitive to the subjective user specified support and confidence thresholds.

Some of these weaknesses are alleviated in (Anandhavalli *et al*., 2010), where a missing item is predicted in four steps. First, they use so-called partitioned-ARM to generate a set of association rules (a ruleset). The next step prunes the ruleset (e.g., by removing redundant rules). From these, rules with the smallest distance from the observed incomplete shopping cart are selected. Finally, the items predicted

by these rules are weighed by the rules' antecedents' similarity to the shopping cart.

The approach proposed by Hewawasam *et al*. (2007) pursues a Dempster-Shafer (DS) belief theoretic approach that accommodates general data imperfections. To reduce the computational burden, Hewawasam *et al*. (2007) employ a data structure called a belief itemset tree. Here too, rule generation is followed by a pruning algorithm that removes redundant rules. In order to predict the missing item, the technique selects a "matching" ruleset-a rule is included in the matching ruleset if the incoming itemset is contained in rule antecedent. If no rules satisfy this condition, then from those rules that have nonempty intersection with the itemset s, rules whose antecedents are "closer" to s according to a given distance criterion (and a user-defined distance threshold) are picked. Confidence of the rule, its "entropy," and the length of its antecedent are used to assign DS theoretic parameters to the rule. Finally, the evidence contained in each rule belonging to the matching rule set is combined or "pooled" via a DS theoretic fusion technique.

In principle, we could adopt any of the above methodologies; but the trouble is that they were all designed primarily for the classification task and not for shopping cart completion. Specifically, the number of times such classifiers have to be invoked would be equal to the number of all distinct items in the database (i.e., n) minus the number of those already present in the shopping cart. This is why we sought to develop a predictor that would predict all items in a computationally tractable manner. Another aspect of these approaches is the enormous amount of effort/cost it takes to obtain a tangible and meaningful set of rules. The root of the problem lies in the apriori-like algorithms used to generate frequent itemsets and the corresponding association rules-the costs become prohibitive when the database is large and complicated. Here, the size and difficulty are determined by four parameters: number of transactions, number of distinct items, average transaction length and the minimum support threshold.

For example, the problem can become intractable if the number of frequent items is large; and whether an item is frequent or not is affected by the minimum support threshold. It is well known that priori-based algorithms suffer from performance degradation in large-scale problems due to combinatorial explosion and repeated passes through the database (Aly *et al*., 2001).

A common problem exists in the recent algorithms is that the constructed data structure cannot be reused in adhoc mining queries or another mining process. The reason for that can either be:

The constructed structure is built after applying some filters on the transaction file and these filters depends on collecting information from the whole database. So the whole database should be checked at least once before constructing the data structure.

The constructed structure changes in each iteration through the execution of the algorithm.

2. MATERIALS AND METHODS

The existing system uses flagged Itemset trees for rule generation purpose. An itemset tree, T, consists of a root and a (possibly empty) set, {T1; . . . ;Tk}, each element of which is an itemset tree. The root is a pair [s, f(s)], where s is an itemset and f(s) is a frequency. If si denotes the itemset associated with the root of the ith subtree, then s is a subset of si; s not equal to si, must be satisfied for all i. The number of nodes in the IT-tree is upper-bounded by twice the number of transactions in the original database.

Note that some of the itemsets in IT-tree (Kubat *et al*., 2003) are identical to at least one of the transactions contained in the original database, whereas others were created during the process of tree building where they came into being as common ancestors of transactions from lower levels. They modified the original tree building algorithm by flagging each node that is identical to at least one transaction. These are indicated by black dots. This is called flagged IT-tree.

Here is an example for an IT-tree. The flagged IT-tree of the database.

D = { [1, 4] , [2, 5] , [1, 2, 3, 4, 5] , [1, 2, 4] , [2, 5] , [2, 4] is as shown in **Fig. 1**.

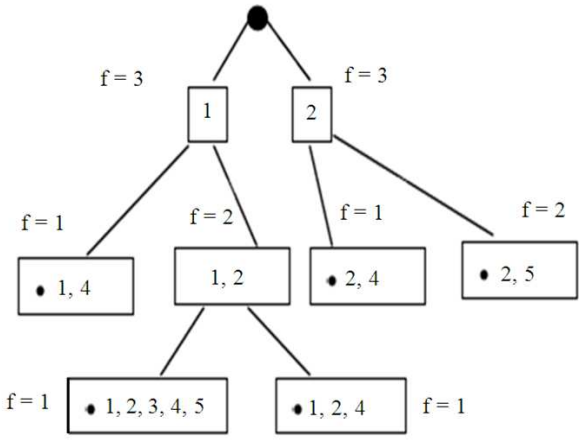

Fig. 1. An example of IT Tree

The following steps are carried out to construct IT tree:

- The first basket is turned into the root, R=[[1,4],1]
- The second basket,[2,5] and S_R=[1,4] are not equal, but have the empty set as a trivial ancestor. a new root, l, is created
- The largest common ancestor of the basket[1,2,3,4,5] and the itemset in the node[[1,4],1] is l=[1]. Therefore, two new nodes are created
- c_i=[1] is an ancestor of the fourth basket,[1,2,4]. This basket is therefore propagated down the subtree rooted at [1]. The recursive call reveals that [1,2,4] has a common ancestor with [1,2,3,4,5] and that the largest such ancestor is [1,2]. A new node is created
- The fifth basket, [2,5], is identical with one of the root's children. The algorithm's only action is to update this node's frequency
- The sixth basket,[2,4], turns out to have a common ancestor with[2,5]. The largest such ancestor, [2], becomes the child of R

2.1. Dempster Combination Rule (DCR)

When searching for a way to predict the presence or absence of an item in partially observed shopping carts, we wanted to use association rules. However, many rules with equal antecedents differ in their consequents-some of these consequents contain those items, others do not. The question is how to combine (and how to quantify) the potentially conflicting evidence. One possibility is to rely on the DS theory of evidence combination (Wickramaratna *et al*., 2009). This technique, which is referred to by the acronym DCR-ARM (Dempster-Combination Rule Association Rule Mining), is described as follows.

Consider a set of mutually exclusive and exhaustive propositions $\Theta = \{\Theta_1 \ldots. \Theta_k\}$, referred to as the Frame of Discernment (FoD). A proposition Θ_i, referred to as a singleton, represents the lowest level of discernible information. In context, Θ_i states that the "value of attribute is equal to Θ_i." Any proposition that is not a singleton, e.g., (Θ_1, Θ_2), is referred to as composite.

An indication of the evidence one has toward all propositions that may themselves imply a given proposition $A \subseteq \Theta$ is quantified via the belief, Bel (A) $\in$ [0,1], defined as:

$$\mathrm{Bel}(A) = \sum_{B \subseteq A} m(B)$$

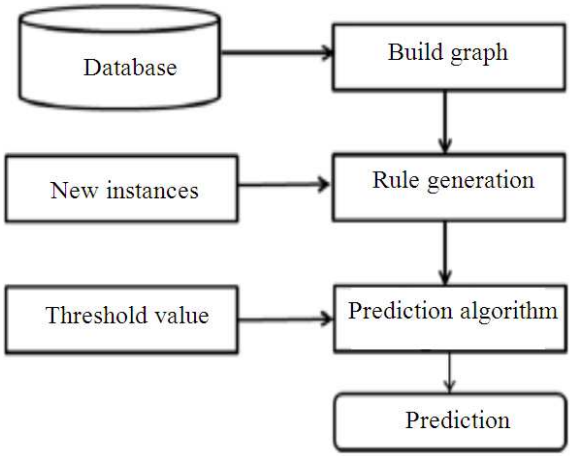

Fig. 2. Shopping cart prediction architecture

Dempster Combination Rule (DCR) is used to combine the discovered association rule. In association mining techniques, a user-set minimum support decides about which rules have "high support." Once the rules are selected, they are all treated the same, irrespective of how high or how low their support. Decisions are then made solely based on the confidence value of the rule. However, a more intuitive approach would give more weight to rules with higher support. Finally the predicted items are suggested to the user

2.2. Proposed Approach

In **Fig.2**, the shopping cart prediction architecture is proposed. Based on passed transaction we can easily construct a Graph structure from which association rules are generated in consideration of new incoming instances in new transaction. Then based on threshold value set by the user and kept dynamic, the prediction algorithm predicts the new item set to be considered for purchase. Threshold value is the minimum support value that a particular pair has to be present before getting predicted.

Proposed Algorithm:

Input: set of items from database
Output: predicted items
// item table generation
for *i*=0: len(*Item*)
IndexMat (*i*, 0) =item (*i*);
IndexMat (*i*, 1) =*key*++;
End
//matrix form of item table

```
for i=0: len(IndexMat)
for j=1: len(index Mat)
AssoMat(i,j)=0;
end
end
//predicting the items
for i=1: len (key Index)
prod=key Index (i);
Pair = AssoMat (prod, prod);
for j=1: len (pair)
Pair Index=pair (j);
edge Value= AssoMat (prod, pair Index);
If edge Value >= Threshold
predict Item=predict Item union pair Index
end
end
end
return predict Item;
//choice of customer
for i=1:len(keyIndex)
prodi=keyIndex(i);
for j=1:len(keyIndex)
prodj=keyIndex (j);
if (prodi==prodj)
AssoMat(prodi,prodj)union =keyIndex
else
AssoMat(prodi,prodj) +=1;
end
end
end
//list of items that the customer has selected with the threshold value
threshold=2;
keyIndex: Null
for k=1:len(item)
for i=1:len (indexMat)
prod=indexMat(i, 1);
if prod==item(k)
keyIndex=keyIndex union indexMat(i, 2);
break;
end
end
end
predict Key=Predict(AssoMat, Threshold, keyIndex)
// display this predict key item as the next item to be purchased
if (choice)
item=item union newSelecteditem from the predictKey
doTransaction(item);
else
update AssoMat (keyIndex)
end
```

In the above algorithm, the input set of items is formulated into an Item table where key values are generated for individual items. Then the Association graph is constructed for every item purchased by the customer. The edge value represents the number of occurrences or frequencies of the items purchased by the same or different customers. At this stage we need the Support value. Then association rules are to be generated from the already generated frequent itemsets. It takes minimum confidence from the user and discovers all rules with a fixed antecedent and with different consequent. The association rules generated form the basis for prediction.

2.3. The Prediction Accuracy Measures

The prediction accuracy need to be evaluated in a situation where several "class attributes" had to be predicted at the same time. For another, plain error rate failed to characterize the predictor's performance in terms of the two fundamental error types: a "false negative," where the predictor incorrectly labeled a positive example of the given class as negative and a "false positive," where the predictor incorrectly labeled a negative example as positive. For these reasons, we preferably relied on the Precision (Pr) and Recall (Re) criteria.

2.4. These Measures are Defined as Follows

Let us denote by TP the number of true positives (correctly labeled positive instances); by FN the number of false negatives; by FP the number of false positives; and by TN the number of true negatives (correctly labeled negative instances). These quantities are used to define Pr and Re in the following way Equation 1 and 2:

$$\text{Precision Pr} = TP / (TP + FP) \quad (1)$$

$$\text{Recall Re} = TP / (TP + FN) \quad (2)$$

The two metrics can be combined together and the so called F_1 measure is defined as Equation 3:

$$F_1 = 2 \text{ X Pr X Re} / (Pr + Re) \quad (3)$$

3. RESULTS

The performance of both the existing tree approach and the proposed approach is analyzed with databases of different sizes. The experiments indicate that the time

needed to answer an average query is approximately linear in the number of market baskets contained in the original database. The time of prediction of the items in our algorithm has decreased to some extent when compared to existing tree approach. More importantly, our approach is very easy to update: Whenever a new set of market baskets become available, they can be incorporated in our algorithm in the graph structure and accordingly the query can be answered.

4. DISCUSSION

The performance of current work to the previous work is compared by considering the attributes like selected items with bread and probable items like jam, butter, juice, milk. The result found is surprising in the current work when the number of transaction also increases.

We experimented with two benchmark domains from the UCI repository that are broadly known, with characteristics well understood by the research community. To be more specific, we used the congressional vote domain and the SPECT-Heart domain. The congressional vote data set has the form of a table where each row represents one congressman or congresswoman and each column represents a bill. The individual fields contain "1" if the person voted in favor of the bill, "0" if he or she voted against and "?" otherwise. We numbered the bills sequentially as they appeared in the table (from left to right) and then converted the data by creating for each politician a "shopping cart" containing the numbers of those (and only those) bills that he or she voted for. For instance, the shopping cart containing (Agrawal and Srikant, 1994; Anandhavalli *et al*., 2010; Kubat *et al*., 2003) indicated that the politician voted for bills represented by the first, fourth and eighth column in the table. In our experiments, we ignored the information about party affiliations (the class label in the original data). The performance of both the existing tree approach and the proposed approach is analyzed with databases of different sizes. The time of prediction of the items in our algorithm has decreased to a considerable extent when compared to existing tree approach.

Apart from congressional vote domain, another binary data set, the SPECT-Heart domain, where 267 instances characterized by 23 Boolean attributes are used. Again, we converted these data into the shopping carts paradigm.

The following **Fig. 3** shows the performance evaluation which compares the performance of both the existing tree approach and proposed approach and displays the time taken to execute for different transactions in seconds. For this experiment, the number of distinct items was 100 and the number of shopping carts in the data set was 1,000. That the costs grew very fast with transaction lengths is to be attributed to the fact that increasing size of shopping carts meant that only a small portion of all shopping carts had empty intersections; the number of generated rules then grew exponentially.

The following **Fig. 4** shows the performance evaluation chart which shows the impact of the growing number of distinct items. Here, we fixed the number of shopping carts to 10,000 and generated synthetic data with varying number of items. For the relatively low (and, hence, expensive) minimum support of 1% and for the number of distinct items varied between 100 and 1,000, we obtained the results shown in **Fig. 4**.

The following **Fig. 5** shows execution time versus minimum support. Even though the rule generation algorithm is not sensitive to minimum support threshold, rule combination costs reduce with increasing minimum support due to reduction in number of rules.

The following **Fig. 6 and 7** shows the performance in Terms of Precision, Recall and F-Value. The performance of the proposed algorithm was good.

The future work may address the various other data structures that will suitably handle large amount of items from the transactional database. The cost of generating the association rule may also be reduced by improved algorithms.

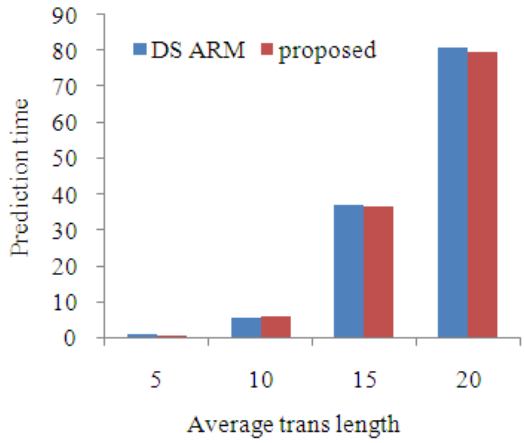

Fig. 3. Average Transaction length Vs Prediction time

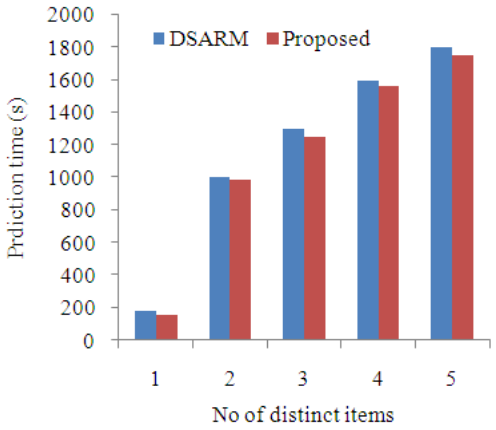

Fig. 4. Distinct items Vs Prediction time

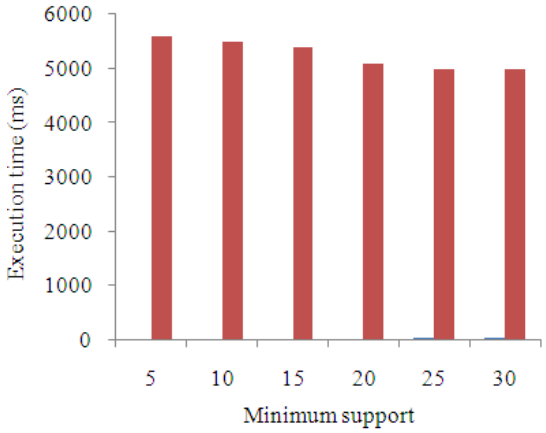

Fig. 5. Execution time Vs Minimum support

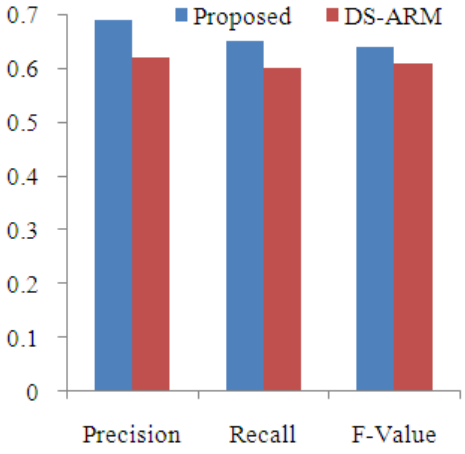

Fig. 6. Precision, Recall and F-Value in the congressional vote domain

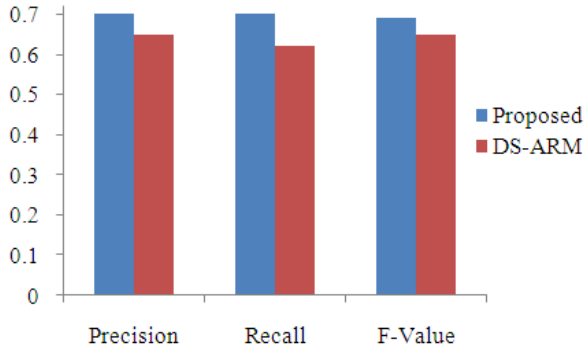

Fig. 7. Precision, Recall and F-Value in the SPECT-Heart domain

5. CONCLUSION

In this study, we have proposed a fast algorithm generating frequent itemsets without generating candidate itemsets. We have shown that the proposed algorithm is simpler and performs well compared to existing approaches especially when more itemset are added to the shopping cart. The execution time is much improved as shown in performance testing. When user adds each item to the cart the algorithm is executed and the prediction is displayed.

The Advantages of our proposed work are:

- Candidate itemsets are not generated
- It uses only a single pass over the database
- The memory consumed is also very less
- Processing speed is more when compared to rules generated using item set tree and DS theory
- High flexibility and user friendly

6. REFERENCES

Agrawal, R. and R. Srikant, 1994. Fast algorithms for mining association rules in large databases. Proceedings of the 20th International Conference on Very Large Data Bases, (VLDB' 94), Morgan Kaufmann Publishers Inc. San Francisco, CA, USA., pp: 487-499.

Aggarwal, C.C., C. Procopius and P.S. Yu, 2002. Finding localized associations in market basket data. IEEE Trans. Knowl. Data Eng., 14: 51-62. DOI: 10.1109/69.979972

Aly, H.H., A.A. Amr and Y. Taha, 2001. Fast mining of association rules in large-scale problems. Proceedings of the 6th IEEE Symposium on Computers and Communications, Jul. 03-05, IEEE Xplore Press, Hammamet, pp: 107-113. DOI: 10.1109/ISCC.2001.935362

Anandhavalli, M., S. Jain, A. Chakraborti, N. Roy and M.K.Ghose, 2010. Mining association rules using fast algorithm. Proceedings of the IEEE 2nd International Advance Computing Conference, Feb. 19-20, IEEE Xplore Press, Patiala, pp: 400-403. DOI: 10.1109/IADCC.2010.5422920

Bollmann-Sdorra, P., A. Hafez and V.V. Raghavan, 2001. A theoretical framework for association mining based on the Boolean retrieval model. Proceedings of the 3rd International Conference on Data Warehousing and Knowledge Discovery, (WKD' 01), Springer-Verlag London, UK., pp: 21-30.

Hewawasam, K.K., G.K. Hewawasam, K. Premaratne and M.L. Shyu, 2007. Rule mining and classification in a situation assessment application: A belief-theoretic approach for handling data imperfections. IEEE Trans. Syst. Man Cybern. B Cybern., 37: 1446-1459. PMID: 18179065

Li, W., J. Han and J. Pei, 2001. CMAR: Accurate and efficient classification based on multiple class-association rules. Proceedings of the IEEE International Conference on Data Mining, Nov. 29-Dec. 02, IEEE Xplore Press, San Jose, CA., pp: 369-376. DOI: 10.1109/ICDM.2001.989541

Liu, J., Y. Pan, K. Wang and J. Han, 2002. Mining frequent item sets by opportunistic projection. Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDDM' 02), pp: 229-238. DOI: 10.1145/775047.775081

Raghavan, V. and A. Hafez, 2000. Dynamic data mining. J. Am. Soc. Inform. Sci.

Kubat, M., A. Hafez, V.V. Raghavan, J.R. Lekkala and W.K. Chen, 2003. Itemset trees for targeted association querying. IEEE Trans. Knowl. Data Eng., 15: 1522-1534. DOI: 10.1109/TKDE.2003.1245290

Wickramaratna, K., M. Kubat and K. Premaratne, 2009. Predicting missing items in shopping carts. IEEE Trans. Knowl. Data Eng., 21: 985-998. DOI: 10.1109/TKDE.2008.229