An E-Commerce application for Presuming Missing Items

Kollipara Anuradha[1], K. Anand Kumar[2]

[1]*Kollipara Anuradha pursuing M.Tech(CSE), Vikas College of Engineering and Technology (VCET), Nunna, Vijayawada. Affiliated to JNTU-Kakinada, A.P, India*

[2] *K. Anand Kumar is working as an Associate Professor in Department of CSE at Vikas College of Engineering and Technology (VCET), Nunna, Vijayawada, India.*

***Abstract*- Data mining is an interdisciplinary subfield of computer science and it is the computational process of discovering patterns in large data sets. The overall goal of the data mining process is to extract information from a dataset and transform it into an understandable structure for further use. Presuming is a technique to assume a future behavior depending on their past behavior. Presuming is a very integral and vital part of Data mining. In E-Commerce application, there are mainly three important parts, first Business Owner, second customer and third Items. Owner earns profit by selling items to customer. It is well known fact from Market-Basket Analysis of Data Mining that Items having associability with each other, and that can be found by mining information from customer buyed products. This analysis helps in finding the item associability. Here we are going to propose a analysis by using which we effectively and easily get the item association and can make the prediction for item which is missing in that transaction. For this we are using Association rule and Apriori Algorithm, in order to reduce the rule mining cost, this algorithm is generating frequent item sets from the transactions which are already done. This algorithm uses Boolean vector with relational AND operation to discover frequent item sets and generate the association rule.**

***Keywords*- Association Rule Mining, Presuming, Data mining, association.**

I-INTRODUCTION

Generally, data mining (sometimes called data or knowledge discovery) is the process of analyzing data from different perspectives and summarizing it into useful information - information that can be used to increase revenue, cuts costs, or both. Data mining software is one of a number of analytical tools for analyzing data. It allows users to analyze data from many different dimensions or angles, categorize it, and summarize the relationships identified. Technically, data mining is the process of finding correlations or patterns among dozens of fields in large relational databases. Most companies collect and refine massive quantities of data. Data mining techniques can be implemented rapidly on existing software and hardware platforms to enhance the value of existing information resources and can be integrated with new products and systems as they are brought on-line. When implemented on high performance client/server or parallel processing computers, data mining tools can analyze massive databases to deliver answers to many questions. The information and knowledge gained can be used for application ranging from market analysis, fraud detection, and customer retention, to production control and science exploration. Data Mining plays an important role in online shopping for analyzing the subscribers□ data and understanding their behaviors and making good decisions such that customer acquisition and customer retention are increased which gives high revenue. Data mining automates the process of finding predictive information in large databases. Questions that traditionally required extensive hands-on analysis can now be answered directly from the data quickly. The primary task of association mining is to detect frequently co-occurring groups of items in transactional databases. The intention is to use this knowledge for prediction purposes.

A. *Association Rule Mining-*

In data mining, association rule learning is a popular and well researched method for discovering interesting relations between variables in large databases. It is intended to identify strong rules discovered in databases using different measures of interestingness. Based on the concept of strong rules, association rules for discovering regularities between products in large-scale transaction data recorded by point-of-sale (POS) systems in supermarkets. For example, the rule {Bread,Butter} =>{Jam}found in the sales data of a supermarket would indicate that if a customer buys onions and potatoes together, he or she is likely to also buy hamburger meat. Such information can be used as the basis for decisions about marketing activities such as, e.g., promotional pricing or product placements.. As opposed to sequence mining, association rule learning typically does not consider the order of items either within a transaction or across transactions.

Following the original definition the problem of association rule mining is defined as: Let $I = \{i1, i2, i3, \ldots, in - 1\}$ be a set of η binary attributes called *items*. Let $D = \{t1, t2, t3, \ldots, tn - 1\}$ be a set of transactions called the *database*. Each transaction in D has a unique transaction ID and contains a subset of the items in I.A *rule* is defined as an implication of the form $X=>Y$ where $X,Y\subseteq I$ and $X\cap Y=.\emptyset$ The sets of items (for short item sets) X and Y are called antecedent (left-hand-side or LHS) and consequent (right-hand-side or RHS) of the rule respectively.

B. Prediction Rule-

Data mining automates the process of finding predictive information in large databases. Questions that traditionally required extensive hands-on analysis can now be answered directly from the data quickly. The primary task of association mining is to detect frequently co-occurring groups of items in transactional databases. The intention is to use this knowledge for prediction purposes. Early attempts for prediction used classification and performance was favorable. In this project, any item is allowed to be treated as a class label its value is to be predicted based on the presence of other items. Put another way, knowing a subset of the shopping carts contents, we want to "guess" (predict) the rest. Suppose the shopping cart of a customer at the checkout counter contains bread, butter, milk, cheese, and pudding. Could someone who met the same customer when the cart contained only bread, butter, and milk, have predicted that the person would add cheese and pudding? It is important to understand that allowing any item to be treated as a class label presents serious challenges as compared with the case of just a single class label.

The number of different items can be very high, perhaps hundreds, or thousand, or even more. To generate association rules for each of them separately would give rise to great many rules with two obvious consequences: first, the memory space occupied by these rules can be many times larger than the original database (because of the task's combinatorial nature); second, identifying the most relevant rules and combining their sometimes conflicting predictions may easily incur prohibitive computational costs. In this work, both of these problems are solved by developing a technique that answers user's queries (for shopping cart completion) in a way that is acceptable not only in terms of accuracy, but also in terms of time and space complexity.

In all these databases, prediction of unknown items can play a very important role. For instance, a patient's symptoms are rarely due to a single cause; two or more diseases usually conspire to make the person sick. Having identified one, the physician tends to focus on how to treat this single disorder, ignoring others that can meanwhile deteriorate the patient's condition. Such unintentional neglect can be prevented by subjecting the patient to all possible lab tests. However, the number of tests one can undergo is limited by such practical factors as time, costs, and the patient's discomfort. A decision support system advising a medical doctor about which other diseases may accompany the ones already diagnosed can help in the selection of the most relevant additional tests.

C.Existing System-

The existing system uses flagged Item set trees for rule generation purpose. An item set tree, T, consists of a root and a (possibly empty) set, $\{T_1; . . . ;T_k\}$, each element of which is an item set tree. The root is a pair [s, f(s)], where s is an item set and f(s) is a frequency. If si denotes the item set associated with the root of the ith sub tree, 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 item sets in IT-tree 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.

But there are some drawbacks in flagged IT-Tree method like Time taken for constructing flagged IT tree is more when compared to Boolean Matrix method and this method requires more memory for processing.

D.Proposed System-

In this proposed system we are going to overcome from the drawbacks of flagged IT tree.Dempster□s rule of combination (DRC) is used to combine the discovered. When searching for a way to predict the presence of an item in partially observed shopping carts, association rules are used. However, many rules with equal antecedents differ in their consequents and some of these consequents contain the desired item to be predicted, others do not. The question is how to combine (and how to quantify) the potentially conflicting evidences. DRC is used for this purpose. Finally the predicted items are suggested to the user.

II-PRESUMING MISSING ITEMS ARCHITECTURE

Presuming missing items architecture having the Boolean Matrix which is generated by transforming the database into Boolean values. The frequent item sets are generated from the Boolean matrix. At this stage we need the Support value. Then association rules are to generated from the already generated frequent item sets. 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. We assign BBA value to each association rule generated. This gives more weight to rules with higher support masses are assigned based on both their confidence and support values. The incoming item set i.e. the content of incoming shopping cart will also be represented by a Boolean vector and AND operation is performed with each transaction vector to generate the association rules. Finally the rules are combined to get the predictions. Demister's rule of combination (DRC) is used to combine the evidences. When searching for a way to predict the presence or absence of an item in a partially observed shopping cart s, we wanted to use association rules. However, many rules with equal antecedents differ in their consequents—some of these consequents contain the desired item to be predicted, others do not. The question is how to combine (and how to quantify) the potentially conflicting evidences. DRC is used for this purpose. Finally the predicted items are suggested to the user.

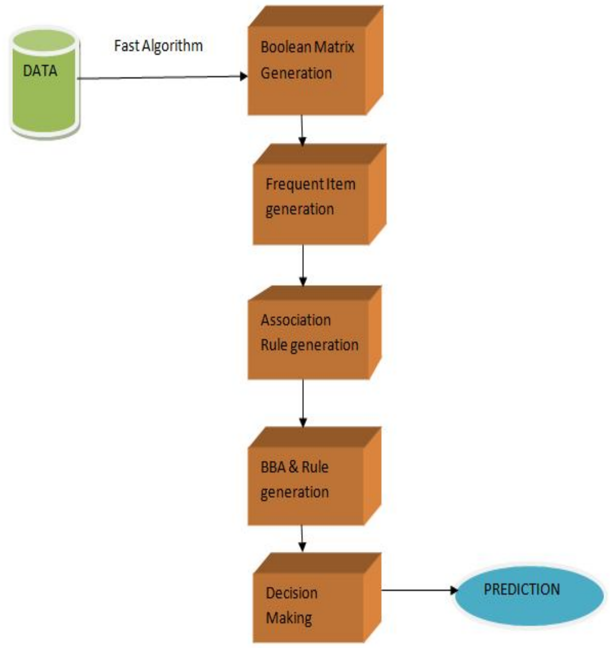

Fig 1-Presuming missing item architecture

III-IMPLEMENTATION

This topic consists of detailed description of each and every module with its advantages and data and execution flow of each module with algorithm. It helps to understand each and every module of the project more deeply and clearly. Each description consists of the basic concept of the module, input and also the excepted output.

A. Boolean Matrix Generation-

This module is to convert the data’s in the database and the incoming instance to database into Boolean value (either 0□s or 1□s). If an item is present in the transaction it is marked with the Boolean value 1 else the item is marked as 0. Raw database “rdb” is a m x n matrix where „m□ is the number of transactions and „n□ is the number of attributes. By using above mentioned rule the Raw database in converted into Boolean database “bdb” (rdb [i, j] => bdb [i, j] where i represent the rows and j represent the columns).

ALGORITHM

```
          for all i<=m do
          for all j<=n do
     if jth item is present in ith row
          set rdb[i,j] =1
          else
          set rdb[i,j] =0
          end do
          end do
```

B. Frequent Item Set Generation-

This module finds out the frequent item set from the existing transaction based on the support value. It involves Join step and Prune step. This module takes the input from the previous stage and forms the frequent item set from matrix table whose values are 1 for transaction. This module also generates the Boolean vector for the frequent item set along with support value. Boolean vector takes the value „true□ for the item present in the item set and takes the value „null□ for the item not present in the item set.

ALGORITHM

```
for each column ci of pdb
if sum(ci) >= new support
f1 = ii
else delete ci from pdb
for each row rj of pdb
if sum(rj) < 2
delete rj from pdb
for (k=2;| fk-1|>k-1;k++)
{
produce k-vectors combination for all columns of
bdb;
for each k-vectors combination {ci1,ci2, ci3 … ,cik }
{
b= ci1 • ci2 •.…•cik
if sum(b)>= new support
fk={ ii1, ii2,……,iik };
}
for each item ii in fk
if | fk(ii)| < k
delete the column ci according to item ii from bdb;
for each row rj from bdb
if sum(rj) < k+1
delete rj from bdb;
k=k+1
}
return f= f1 u f2 … u fk
```

C. Association Rule Generation-

This module is used to generate association rules from the already generated frequent item sets. The algorithm uses the fact that:

"If there exists two rules A->C and A->{C U X} where X doesn't belongs to A U C then the confidence of the second cannot be larger than the first one".

The algorithm checks if a given set is a subset of another set or not. To perform this operation each item in an item set is represented as an integer where a bit corresponding to as item is set to 1.

This algorithm is capable of finding all association rules with a fixed antecedent and with different consequents from the frequent item sets subject to a user specified minimum confidence very quickly. It takes minimum confidence from the user and discovers all rules with a fixed antecedent and with different consequent. This module also takes the frequent item set and the incoming shopping cart instance to generate the association rule with the corresponding support and confidence value.

ALGORITHM

```
for all fk, fk ∈ F, 1<=k<=maxsize-1 do begin
Rsup=support (fk)*miconf
found=0
for all fm, fm _ Fk +1<= m <=maxsize do begin
 If (support(fm)>=rsup) then begin
 If (fk ⊂ fm) then begin
  Found=Found+1
 Conf=support (fm)/ support (fk)
Generate the rule fk = (fm - fk) &= conf and
Support=support (fm)
End if
Else
If (found<2)
Continue step1 with next k
Else found=0
Endif
Endif
End do
End do
```

D. BBA and Decision Making-

a).PARTITIONED SUPPORT-

The partitioned-support p_supp of the rule, r (a) -> r(c), is the percentage of transactions that contain r (a) among those transactions that contain r(c), i.e.,

$$\mathbf{p\_supp = support(r\ (a)\ U\ r(c))\ /\ support(r(c))}$$

b)Basic Belief Assignment(BBA)-

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. Therefore, we use a novel method to assign to the rules masses based on both their confidence and support values. This weight value is called Basic Belief Assignment (BBA)

We assign BBA value to each association rule generated.

$$\boldsymbol{\beta = ((1+\alpha 2)\ x\ conf\ x\ p\_supp)\ /\ (\alpha 2\ x\ conf + p\_supp);} \qquad \text{where } \alpha \text{ € } [0,1];$$

E .Demister's Rule-

Demister's rule of combination (DRC) is used to combine the evidences. 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 the desired item to be predicted, others do not. The question is how to combine (and how to quantify) the potentially conflicting evidences. DRC is used for this purpose.

IV- ANALYSIS

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

NO.OF TRANSACTION	TREE APPROACH (Execution time)	PROPOSED APPROACH (Execution time)
100	48.7	0.797
60	42.031	0.578
20	38.721	0.547

Fig 2-Execution time comparison

V-PERFORMANCE EVALUATION

The Fig.3. Shows the performance evaluation graph 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.

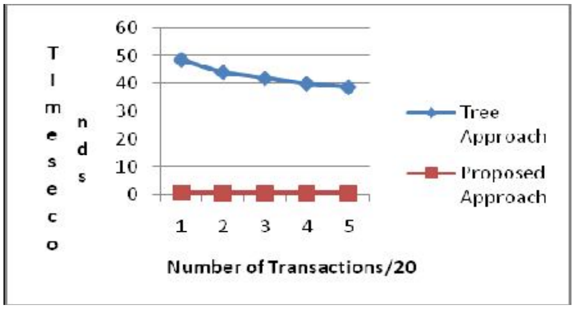

Fig 3-Performance Evaluation Graph

VI- CONCLUSION

This proposed algorithm for presuming missing items in e-commerce business is effective and useful for both the user and shopping site. This proposed system is able to generate frequent item sets .The performance of the proposed system is much better and well applicable compare to existing approach. This algorithm uses Boolean vector and relational AND operation to find frequent item sets and by use of frequent item sets algorithm generates Association Rules. Association rule is basis for prediction. Then by using Basic Belief Assignment and Decision Rule technique we are able to assume missing item in short period of time compare to existing approach. This algorithm has been applied to a demo shopping application, when user adds items to its shopping cart, the algorithm is executed and prediction is displayed.

VII-REFFERENCES

R. Agrawal and R. Spirant, “Fast Algorithms for Mining Association Rules,” Proc. Intel Conf.Very Large Databases(VLDB 94), pp.487-499, 1994.

K.K.R.G.K. Hewawasam, K. Premaratne, and M.-L. Shyu, “Rule Mining and Classification in a Situation Assessment Application: A Belief Theoretic Approach for Handling

Data Imperfections,” IEEE Trans. Systems, Man, Cybernetics, B, vol. 37, no. 6 pp. 1446-1459, Dec. 2007. Appriori Algorithm Reference

P. Bollmann-Sdorra, A. Hafez, and V.V. Raghavan, “A Theoretical Framework for Association Mining Based on the Boolean Retrieval Model,” Data Warehousing and Knowledge Discovery.

Kansu Wickramaratna, Miroslav Kubat and Kamal Premaratne, “Predicting Missing Items in Shopping Carts”, IEEE Trans.

VII-AUTHORS PROFILE

Kollipara Anuradha, Pursuing M.Tech(CSE) Vikas College of Engineering and Technology (VCET), Nunna, Vijayawada. Affiliated to JNTU-Kakinada, A.P., India

K. Anand Kumar, is working as a Associate Professor in CSE department at Vikas College of Engineering and Technology(VCET), Nunna, Vijayawada, Affiliated to JNTU-Kakinada, A.P., India