Predicting Missing Items in Shopping Cart using Associative Classification Mining

Ila Padhi
Department of IT
College of Engineering and Technology
Bhubaneswar, Odisha, India

Jibitesh Mishra
Department of IT
College of Engineering and Technology
Bhubaneswar, Odisha, India

Sanjit Kumar Dash
Department of IT
College of Engineering and Technology
Bhubaneswar, Odisha, India

ABSTRACT

The primary task of association rule mining is to detect frequently co-occurring groups of items in transactional databases. The intention is to use this knowledge for prediction purposes. So 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 paper contributes to the latter 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, for example, If bread, butter, and milk often appear in the same item, then the presence of butter and milk in a shopping cart suggests that the customer may also buy bread. More generally knowing which items a shopping cart contains, we want to predict often items that the customer is likely to add before proceeding to the checkouts. So this paper presents a technique called the "Combo Matrix" 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 currently 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 Combo Matrix. Then association rules are to be generated from the already generated frequent itemsets. The association rules generated form the basis for prediction. The incoming itemsets i.e. the contents of the shopping cart will be represented by set of unique indexed numbers and the association among items is generated through the Combo Matrix. Finally the predicted items are suggested to the Customer.

Keywords

Association rule mining, Prediction, Frequent Item set, Combo Matrix, Incidence Matrix

1. INTRODUCTION

Data mining, “the extraction of hidden predictive information from large databases”, is a powerful new technology with great potential to help companies focus on the most important information in their data warehouses. Data mining tools predict future trends and behaviors, allowing business to make proactive, knowledge-driven databases.

Association Rule Mining [1], [4], [5], [6] is a popular and well researched method for discovering interesting relations between variables in large databases. Association rules are statements of the form *{X1, X2, ..., Xn}⇒Y* meaning that if all of *X1, X2,… Xn* is found in the market basket, and then we have good chance of finding Y. the probability of finding Y for us to accept this rule is called the confidence of the rule. Normally rules that have a confidence above a certain threshold only will be searched. In many situations, association rules involves sets of items that appear frequently. 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 primary task of association mining is to predict 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 [6], [7] 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 cart’s contents, we want to “guess” (predict) [3] the rest. Suppose the shopping cart of a customer at the checkout counter contains bread, butter, milk, cheese, and pudding. But when the customer finally makes her checkout it is found that cheese and pudding are the two items that has been added. So 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 in terms of time and space complexity.

2. RELATED WORKS

Prediction of missing items [2], [3] uses the concept of flagged itemset trees (IT-TREE) for rule generation purpose. An itemset tree, T, consists of a root and a (possibly empty) set, {T_1, T_2,…., 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 s_i denotes the itemset associated with the root of the i^{th} sub tree, then s is a subset of s_i; s not equal to s_i, 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 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. The authors 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 this methodology comes to a complex scenario when number of items increases leads to difficulty in maintaining the IT-TREE.

The Graph based algorithm proposed [8], [9] efficiently solves the problem of mining association rules. Both the algorithms outperform the previous algorithm by scanning the database only once and also producing few candidates. A Graph can be drawn using large itemsets where each itesmset is randomly numbered and stored in database in from of bit vectors. A bitvector represents a transaction where 1 represents presence of an item and 0 represents absence of an item.

3. PROPOSED METHODOLOGY

Here we devise a novel rule generation methodology to limit the number of generated rules while still ensuring that all the interesting rules are discovered. The general concept of association rule mining and special case of mining a relationship that corresponds to a graph. Let X, Y, A, B, C be the set of item. Let X and Y are the two products that have been purchased N number of times together in different transaction by different customer. Then there must be relationship exist in terms of graph in between X and Y, as shown in figure 1, where A, B, C are isolated indicating that in no transaction they have occurred once.

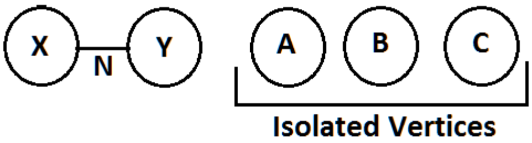


Fig 1: Association graph between item X and Y

Now suppose there is another transaction which consists of itemset A,B,Y then the graph will look as fig 2 where edge value 1 presents that this pair are present once in any transaction. It can be argued that the transformation of graph from 1 to 2 will cost high computation. But it can be considered as that all the items are selected as vertices and are isolated from each other, unless a transaction has occurred where any items are selected. Once item is selected, there should be edge present between those itemset and edge value should be incremented to represent how many times the item pair has been in different transactions. Now let someone purchase Y then he/she has got higher probability to purchase X, A, B but not C as these vertices are connected with each other but not C.

This graph based association not only predicts the next items to be bought but also present the details of items and number of times the items have been bought. So in the next transaction, it not only helps the customer to predict the next item but also the retailers to know which item pair is bought maximum times by the customer. One approach that data structure gives in support of presenting graph in memory is Incidence matrix. We do the same by storing the information about association graph in incidence matrix and the principal diagonal elements represents the associations related to any item. Hence we name this matrix as Combo Matrix**.** So the graph in fig 2 can be represented as table 1.

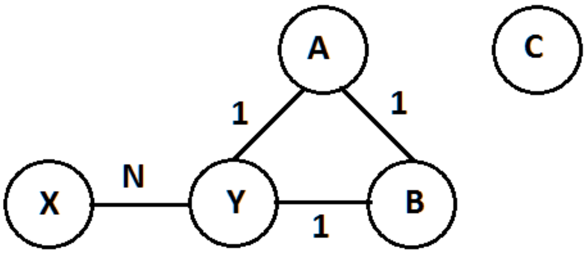


Fig 2: Association graph for item X, Y, A, B, C

Table 1. Combo Matrix for association Graph

Items	**X**	**Y**	**A**	**B**	**C**
X	Y	N	0	0	0
Y	N	A,B,X	1	1	0
A	0	1	Y,B	1	0
B	0	1	1	A,Y	0
C	0	0	0	0	0

In the Combo Matrix vertex ($V_{i,}$ Vi) contains the information about the vertices which has been selected with V_{i}. Let's check how our matrix supports prediction. Let the customer selects Y then looking to the matrix for vertex(Y, Y) it can be assumed that the customer can go for probable items like A, B, X also. But in which order the probable items should be displayed. So the individual for value of vertex(Y, B), vertex(Y, X) as 1, 1, N. Now after sorting we get N, 1, 1 which suggests that item X is having higher probability than A, B.

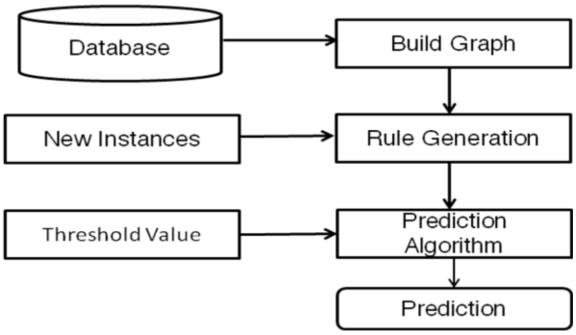


Fig 3: Architecture of the Proposed Work

In Fig.3, based on passed transaction one can easily create a Graph (Combo Matrix) from which rules are generated in consideration of new oncoming items pair 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 range that a pair has to be present before getting predicted.

For example suppose in any transaction customer has selected Y item, then from the graph as shown in table 3, we can generate the rule that the customer can buy item X, A, B .Now let's say the threshold value is 2. For prediction, any pair of item there must be edge value greater than 2, that is item set X and Y has been referred more than twice in different Transaction. Therefore from the rule that for the purchase of item Y, prediction algorithm discard item A,B(as the edge value is less than threshold) and only predict item X to be the next item that the customer is likely to buy. Once the transaction is over, database is updated so it can be reflected in next transaction. The database is blank before initiating a business as there is no transaction that has been made ever. So customer choice is not present, hence the algorithm cannot predict. Once transaction keeps on increasing, prediction logic starts working.

4. ALGORITHMS

4.1 Algorithm to Generate Key Item

```
Item:  is list of item to be sold in any business.
Key=1; // key that to be initialized to each item
Index Mat: is a matrix that stores name of item and Unique key.
for i=1: length(Item)
        IndexMat (i, 1) =item (i);
        IndexMat (i, 2) =key++;
end
```

Table 2. IndexMat which represents unique integer assigned to each item

NAME	KEY
MILK	1
BREAD	2
BUTTER	3
BANANA	4
APPLE	5

Table 2 represents with each item is assigned a unique key value which is all stored in a matrix called Indexmat For example, the items like (MILK, BREAD, BANANA, APPLE, BUTTER) are assigned with a unique key value which is stored in table 2

4.2 Initialization of Combo Matrix

```
InitializeComboMat (IndexMat)
Combo Mat: is a matrix that will represent our association graph
for i=1: length (IndexMat)
    for  j=1: length(index Mat)
        ComboMat(i,j)=0;
    end
end
```

Table 3. Initialization of the Combo Matrix

keyIndex	1	2	3	4	5
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0
5	0	0	0	0	0

The next task is to initialize the Graph. Each vertex is supposed to be isolated from one another. This is done by initializing the Combo Matrix as zero. Representing each edge is having zero value. This is done in Algorithm 4.2 whose output is shown in table 3

4.3 Prediction of Items

```
predict (ComboMat, Threshold, key Index)
Threshold: is the minimum value that is required by pair
Key Index: list of item unique key that are purchased by the customer
predict Item: is an array of item to be predicted, initially it contain NULL
prod: will contain each item key Index that is chosen by customer in new transaction list.
pair: will contain list of item that will be present in diagonal element of Combo matrix of prod.
pair Index: will contain individual item key index of each pair.
edge Value: used for storing the edge value between different pair.
for i=1: length (key Index)
  prod=key Index (i);
  Pair = ComboMat (prod, prod);
    for j=1: length (pair)
     Pair Index=pair (j);
     edge Value= ComboMat (prod, pair Index);
         If edge Value >= Threshold
           predict Item=predict Item ∪ pair Index
         end
     end
end
return predict Item;
```

Algorithm 4.3 returns all the predict items that are greater than the threshold value and which matches our rule.

4.4 Update the Combo Matrix

```
UpdateComboMat (key Index)
keyIndex: list of item keyIndex that the customer has purchased.
for i=1:length(keyIndex)
  prod_i=keyIndex(i);
   for j=1:length(keyIndex)
     prodj=keyIndex (j);
      if (prod_i==prod_j)
        ComboMat(prod_i,prod_j)∪=keyIndex
      else
        ComboMat(prod_i,prod_j) +=1;
      end
   end
end
```

Algorithm 4.4 is invoked either there is any updation of customer’s choice.

4.5 Transaction Occurrences

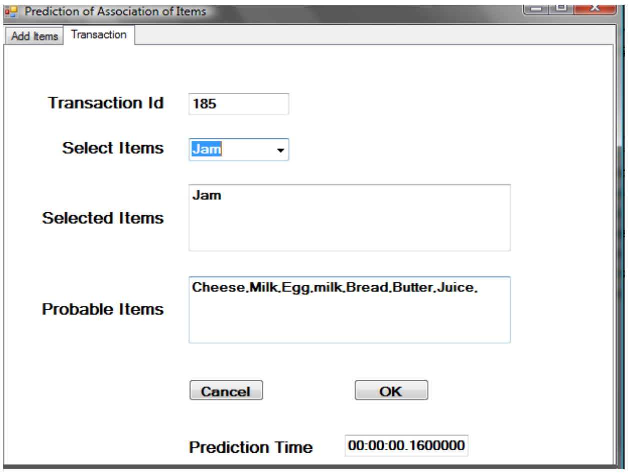
```
do Transaction (item)
item: list of item that customer has selected
threshold=2;
keyIndex: Null
for k=1:length(item)
for i=1:length (indexMat)
   prod=indexMat(i, 1);
     if prod==item(k)
       keyIndex=keyIndex∪indexMat(i, 2);
       break;
     end
end
end
predict Key=Predict(ComboMat, Threshold, keyIndex)
// display this predict key item as the next item to be
```

```
purchased
if (choice)
  item=item ∪ newSelecteditem from the predictKey
  doTransaction(item);
else
  updateComboMat (keyIndex)
end
return;
```

After initializing the Combo Matrix, transaction phase starts. With each new transaction occurs, algorithm DO TRANSACTION is invoked The DO TRANSACTION algorithm is invoked each time whenever the customer selects some items. So first of all the key index of each item is generated using loops. Again it invokes PREDICT algorithm to predict the next items that the customer can go for a selection given a set of items. If the customer makes any selection then the selection of item list gets updated and a recursive call is made to DO TRANSACTION algorithm. Otherwise if the customer does not make any selection then the transaction is updated by invoking the UPDATE COMBO MATRIX.

5. PERFORMANCE EVALUATIONS

The algorithms mentioned in the above section have been evaluated by developing a predictor and the same is compared with the previous predictors for predicting missing items in a shopping cart. Fig. 4 represents previous work i.e. the selected item and the probable items made by the customer along with the prediction time



Transaction Id: 185

Select Items: Jam

Selected Items: Jam

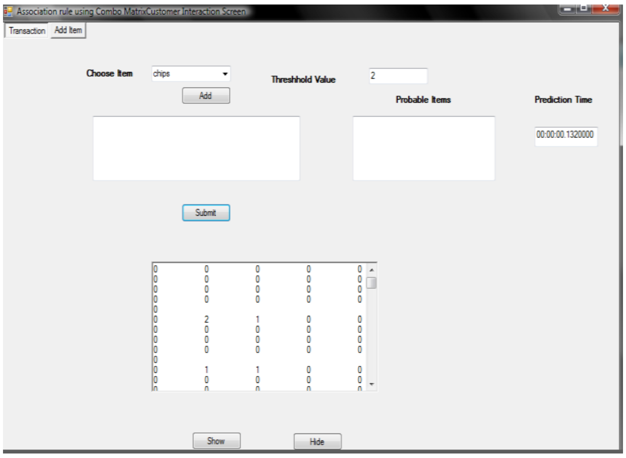
Probable Items: Cheese,Milk,Egg,milk,Bread,Butter,Juice.

Prediction Time: 00:00:00.1600000

Fig 4: Snapshot of Previous work for Prediction

Fig 4 represents the previous work in which an instance of a transaction for example if a customer selects the Jam then the probable items like cheese, milk, egg, bread, butter and juice will be displayed along with the prediction time for that particular transaction.

The current work in fig.5 takes the items from the customer and will only work if the threshold value is 2.There is a facility of insertion of items in the add item screen and the respective items will be automatically assigned unique number by the system. Display of database has been under the control of customer, if the customer wants that the database should be displayed then clicking the Display button will help the necessary actions to be done. In the transaction screen, a pop window will be displayed if the customer enters less number of times than specified in the threshold box. Threshold value is kept dynamic in order to reduce the prediction time and after that the probable items will be displayed in that period of transactions made by the customer time to time. It will only show the prediction of items of the current transaction at different instances. When the customer submits the items by clicking the button, then the prediction time for that particular transaction is displayed. The Combo Matrix displayed at the bottom of the figure indicates the values representing the items in rows and columns. The diagonal elements represent the association among items and the corresponding rows and columns will get updated after each transaction. So ultimately looking to the Combo Matrix, one can predict how many times an item has been bought along with the association of that items and the frequent times of that particular item that has been chosen with other items.


Association rule using Combo MatrixCustomer Interaction Screen
Transaction
Add Item
Add
Probable Items
Submit
Show
Hide

Choose Item: chips

Threshhold Value: 2

Prediction Time: 00:00:00.1320000

```
0    0    0    0    0
0    0    0    0    0
0    0    0    0    0
0    0    0    0    0
0
0    2    1    0    0
0    0    0    0    0
0    0    0    0    0
0    0    0    0    0
0
0    1    1    0    0
0    0    0    0    0
```

Fig 5: Snapshot current work selection of items in Combo Matrix along with the prediction time.

The performance of current work to the previous work is compared by considering the attributes like selected items with Jam and probable items like bread, butter, juice, milk. The result found is surprising in the current work when the number of transaction increases. The time of prediction in the previous work is 0.16000 ms and the time of prediction in current work is 0.13200ms.

6. CONCLUSIONS

We studied the efficient graph based algorithm for transactional database. It efficiently solves the problem of mining association rules by improving the execution time than the previous algorithm as shown in the performance testing. The algorithm uses the Combo matrix to generate the frequent itemsets and the prediction of items. The advantages of the proposed work are:

- It does not generate candidate itemsets.
- It uses only a single pass over the database.
- Memory consumption is less
- Processing speed is more due to use of Combo Matrix.
- More flexible and user Friendly for customers

7. REFERENCES

[1] Agrawal. R, Imielinski. T and Swami. A, "Mining Association Rules between Sets of Items in Large Databases," Proc. ACM Special Interest Group on Management of Data (ACM SIGMOD), pp. 207-216, 1993.

[2] M. Kubat, Hafez. A, Raghavan V.V, J.R. Lekkala and hen W.K.: "Itemset Trees for Targeted Association Querying," IEEE Trans .Knowledge and Data Eng., vol. 15, no. 6, pp. 1522-1534, Nov./Dec.2003.

[3] Kasun Wickramaratna ," Predicting missing Items

in Shopping Carts", IEEE Transaction on Knowledge and data engineering, Vol. 21, No. 7, July 2009

[4] C.C Aggarwal, C. Procopius, and P.S. Yu, "Finding Localized Associations in Market Basket Data," IEEE Transaction on Knowledge and Data": Eng., vol. 14, no. 1, pp. 51-62, Jan./Feb. 2002.

[5] R. Bayardo and R. Agrawal, "Mining the Most Interesting Rules," Proc. ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining, pp. 145-154, 1999.

[6] Liu. B, Hsu, Y.M. WAND Ma,: "Integrating Classification and Association Rule Mining," Proc. ACM SIGKDD Int'l Conf. Know. Disc. Data. Mining (KDD '98), pp. 80-86, Aug. 1998.

[7] Li .W, Han. J, and Pei. J, "CMAR: Accurate and Efficient Classification Based on Multiple Class-Association Rules," Proc. IEEE Int'l Conf. Data Mining (ICDM '01), pp.369-376, Nov./Dec. 2001

[8] Yen S.J. and Chen A " An efficient approach to discovering knowledge from large database". In proc. Of the IEEE/ACM International Conference on parallel distributed Information system, Pages 8-18, 1996.

[9] Lee K.L, G Lee and Chen A.L.P. Efficient Graph based Algorithm for discovering and maintaining association rules in large database.