A Radical Approach for Market Basket Analysis under the Framework of Binary Transaction based improved Apriori Algorithm

Abhijit Sarkar, Assistant Professor
Apurba Paul, Assistant Professor
Anupam Mondal, Assistant Professor
Subhadip Nandi, M.Tech student
Dept. of CSE
JIS College of Engineering
Kalyani, Nadia,
West Bengal,
India

ABSTRACT
In modern world, analyzing data and extracting useful information from the data is one of the crucial task in business analysis. Now, extracting patterns from the data has occurred from centuries. Bayes theorem (used in the 1700s), Regression analysis (used in the 1800s) were the earlier process of identifying and extracting pattern from a huge collection of relevant data. In this regard, association rule learning is one of the popular, well researched procedures to extract pattern or rules to gather information from huge relevant data. In this paper, a new approach of segregating data and generating rules under the framework of binary transaction based modified enhanced Apriori Algorithm have been presented. This algorithm can be applied in any number of data efficiently. Apart from Market Basket Analysis, this algorithm can be applied in the field of web usage mining, intrusion detection, bioinformatics etc.

Keywords
Apriori algorithm, association rules, itemset, binary transaction, supports count.

1. INTRODUCTION

Data mining is the process of extracting useful patterns for business analysis from huge amount of data.[1] Now, data mining helps us to transform raw data into business acumen. In data mining, the concept of association rule learning can be used to forecast future trends and behaviors in business, drilling down into transactional database generating useful rules or patterns. In the traditional Apriori algorithm, we find out the frequent1-itemset, frequent 2-itemset and so on to originate the items which can generate association rules. After originating these items, we calculate confidence of different association rules comprising these frequent itemset existing in association rule set and filter them based on minimum confidence to find out desired rules that helps us in developing marketing decisions. The foremost limitation of this is that we get huge number of tables, and we require huge amount of space to store these tables, if the number of data is high. Because of this, there subsists frequent scanning of massive data in the processing. [2]
In this paper, a modified Apriori Algorithm based on binary transaction have been presented. There will be easy and less access to the database which is proficient in comparison to the conventional approach.

2. APRIORI ALGORITHM
1. k = 1
2. Find frequent set L_k from C_k of all candidate itemsets
3. Form C_{k+1} from L_k; k = k + 1
4. Repeat 2-3 until C_k is empty

- Details about steps 2 and 3

Step 2: scan *D* and count each itemset in C_k, if it's greater than minSup, it is frequent

Step 3:

- For k=1, C_1 = all 1-itemsets.
- For k>1, generate C_k from L_{k-1} as follows:
  - *The join step*

  C_k = join of L_{k-1} with itself (itemset of size k)

If both $\{a_1,\ldots,a_{k-2}, a_{k-1}\}$ & $\{a_1,\ldots, a_{k-2}, a_k\}$ are in L_{k-1}, then add $\{a_1,\ldots,a_{k-2}, a_{k-1}, a_k\}$ to C_k

(We keep items **sorted**).

- *The* prune *step*

Remove $\{a_1,\ldots,a_{k-2}, a_{k-1}, a_k\}$ if it contains a non-frequent (k-1) subset
[1]

3. ILLUSTRATION OF APRIORI ALGORITHM

TID	List of items
T1	I1, I2, I3, I5
T2	I2, I3, I4
T3	I1, I3, I4
T4	I1, I2, I5
T5	I1, I2, I3, I5
T6	I1, I5
T7	I1, I4, I5
T8	I2, I3, I4

Figure 1: Transaction database

Figure 1 shows the transaction database. It contains eight different transactions and the items purchased within the transaction.

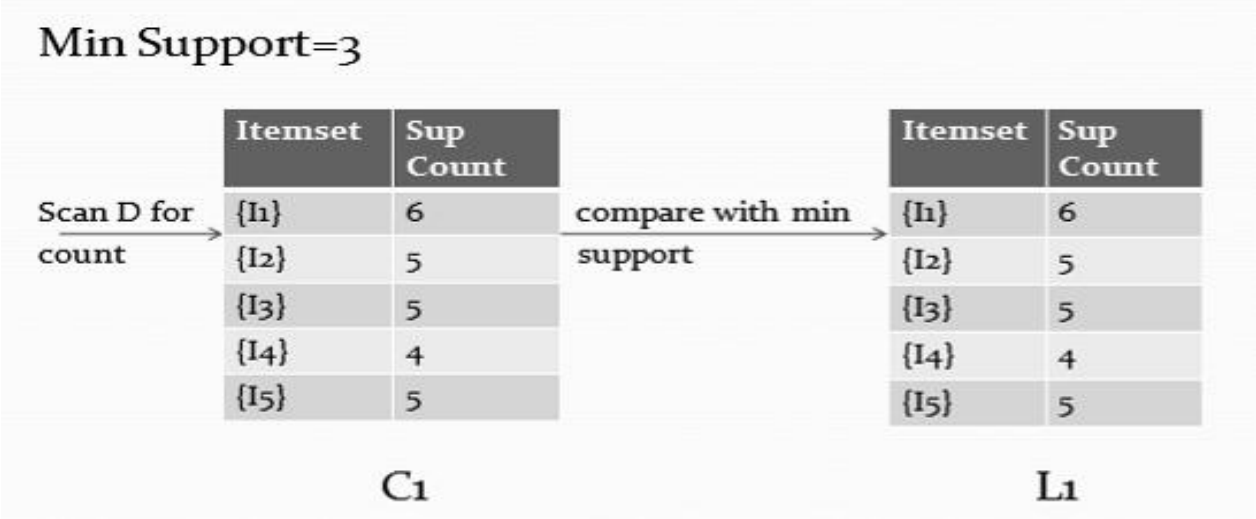

Figure 2: Generation of candidate itemset and frequent 1-itemset

Figure 2 shows the support count of different 1-itemsets as a result of counting the number of appearance in the transaction database of Figure 1 and the selected frequent 1-itemsets based on minimum support threshold value 3 after pruning the item whose support count is less than the threshold value which is 3 in this example.
.

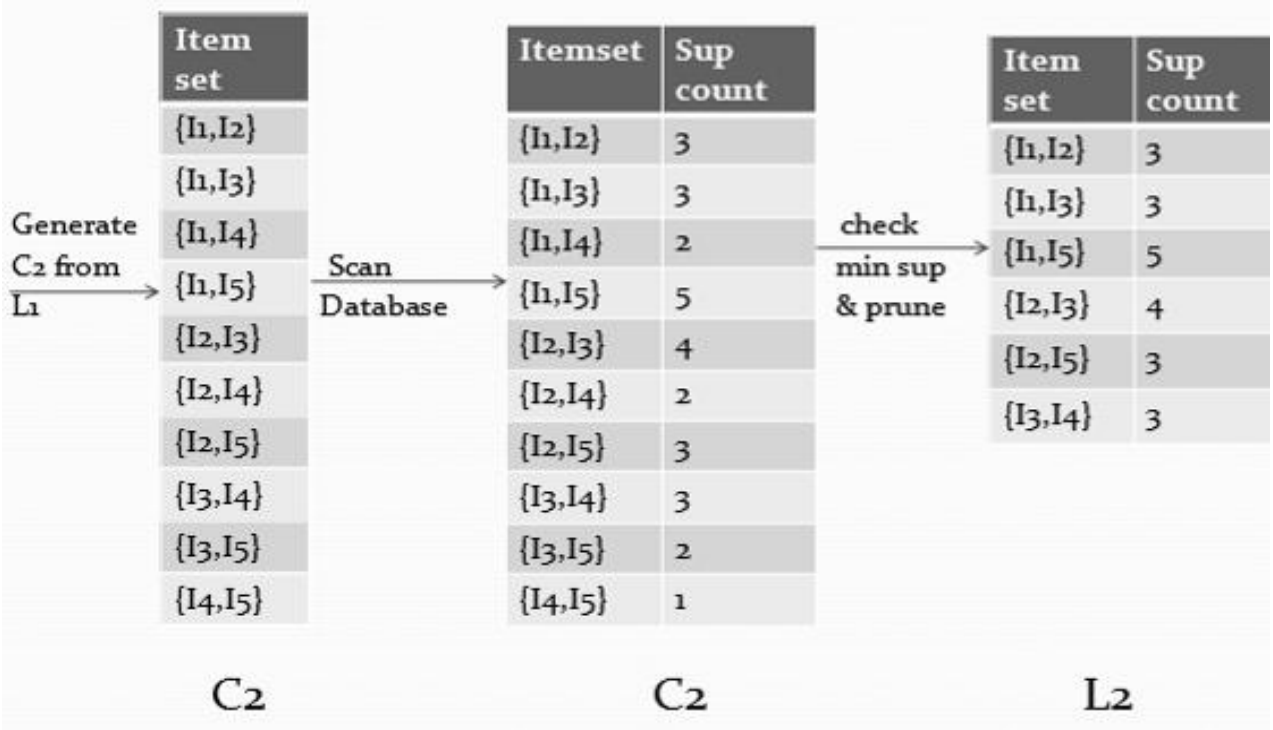

Figure 3: Generation of candidate itemset and frequent 2-itemset

Figure 3 shows the support count of different candidate 2-itemsets as a result of counting the number of appearance in the transaction database of Figure 1 and the selected frequent 2-itemsets based on minimum support threshold value 3 after pruning the item whose support count is less than the threshold value which is 3 in this example.

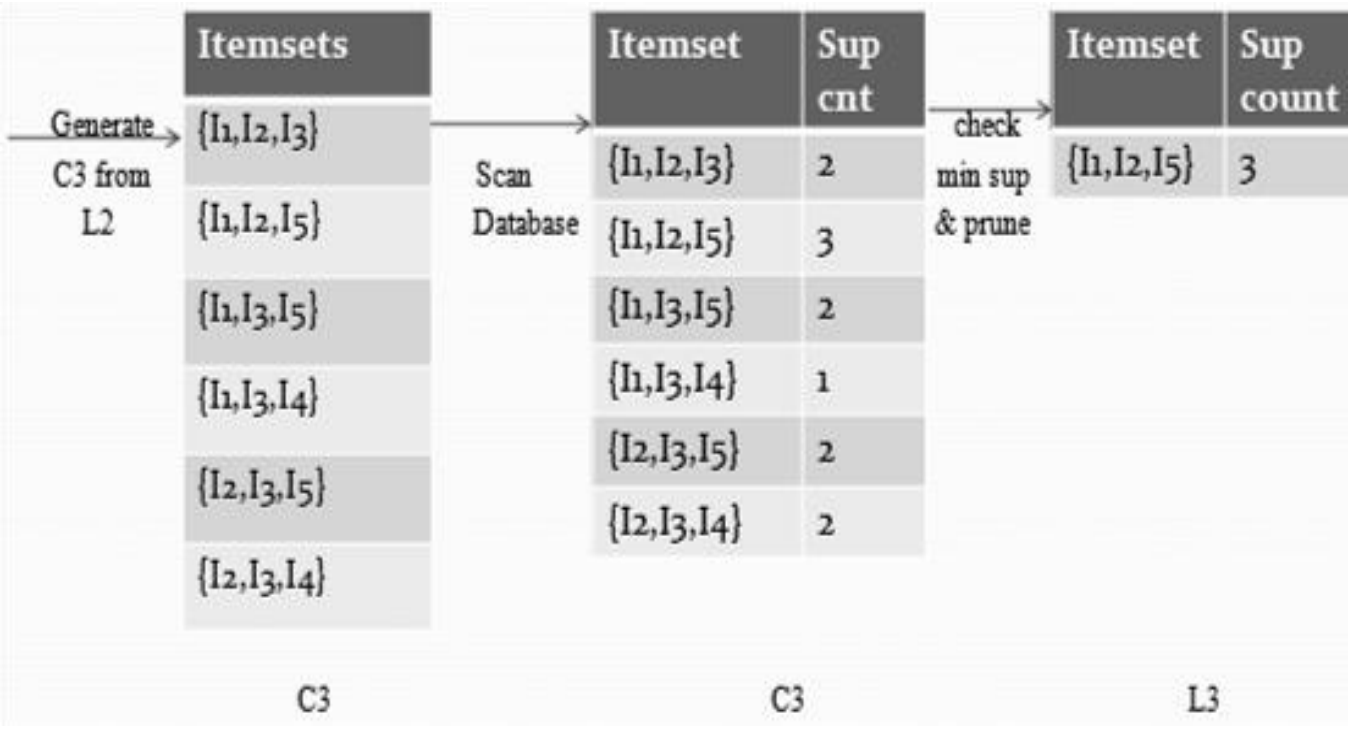

Figure 4: Generation of candidate 3-itemset and frequent 3-itemset

Figure 4 shows the support count of different candidate 3-itemsets and the selected frequent 3-itemset based on minimum support threshold value 3 after pruning and termination of frequent itemset generation. The association rule set which is closing frequent itemset contains $\{I_1, I_2, I_5\}$.

Association rules	Confidence	Status
R1: I1^I2-> I5	Sc{I1,I2,I5}/Sc{I1^I2}=3/3=100%	Selected
R2: I2^I5->I1	Sc{I1,I2,I5}/Sc{I2,I5}=3/3=100%	Selected
R3: I1^I5->I2	Sc{I1,I2,I5}/Sc{I1,I5}=3/5=60%	Rejected
R4: I1->I2^I5	Sc{I1,I2,I5}/Sc{I1}=3/6=50%	Rejected
R5: I2->I1^I5	Sc{I1,I2,I5}/Sc{I2}=3/5=60%	Rejected
R6: I5->I1^I2	Sc{I1,I2,I5}/Sc{I5}=3/5=60%	Rejected

Figure 5: List of selected association rules based on confidence threshold = 70%

Figure 5 shows the different possible association rules from the item existing in association rule set and selection of association rules R1 and R2 among them based on minimum confidence threshold = 70%.

4. BINARY TRANSACTION BASED APRIORI ALGORITHM

Step 1: All the transaction's product item is considered and tabulated in columns to form a transaction table.

Step 2: For a particular transaction, if one product/item is present then under that item's column binary one (1) is written, else if the item is absent then binary zero (0) is written.

Step 3: ARB flag is set to zero (0) initially and Rule Set (containing items which can form association rule) is initially empty.

Step 4: Now, number of 1's for all the items is counted column-wise. Total number of 1's present under each item is measured and item(s) with highest count is considered. If the count is less than Minimum Support then go to Step 7.

Step 5: If the number of item having highest count is 1, then that item is selected and inserted in Rule Set. In the selected item column, that column & the corresponding row(s) with zero (0) in that column is deleted from the transaction table and the table is updated; go to Step 4.

Step 6: If the number of item having highest count is more than one, then ARB is set to 1 and any arbitrary item is selected and inserted in Rule Set. In the selected item column, that column & the corresponding row(s) with zero (0) in that column is deleted from the transaction table and the table is updated; go to Step 4.

Step 7: If ARB=1, then the iteration in which it was made 1 is considered. At that iteration, each other item(s) having same highest count is considered (if not included in Rule Set) instead of selected arbitrary item and repeat from Step 5. If ARB=0, then the algorithm terminates. Rule Set gives the items forming association rules.

5. BINARY TRANSACTION BASED APRIORI ALGORITHM ILLUSTRATION

TID	List of items
T1	I1, I2, I3, I5
T2	I2, I3, I4
T3	I1, I3, I4
T4	I1, I2, I5
T5	I1, I2, I3, I5
T6	I1, I5
T7	I1, I4, I5
T8	I2, I3, I4

Figure 6: Transaction database

Fig 6 shows the transaction database. It contains eight different transactions and the items ID which have been purchased (same as Figure 1).

Transaction	I1	I2	I3	I4	I5
T1	1	1	1	0	1
~~T2~~	~~0~~	~~1~~	~~1~~	~~1~~	~~0~~
T3	1	0	1	1	0
T4	1	1	0	0	1
T5	1	1	1	0	1
T6	1	0	0	0	1
T7	1	0	0	1	1
~~T8~~	~~0~~	~~1~~	~~1~~	~~1~~	~~0~~
COUNT	6>=Min Sup	5	5	4	5

Min. Support=3

Rule Set= {I1}

ARB=0

Figure 7: Finding out first item to be selected in Rule Set Based on highest Support Count from the binary transaction database

Figure 7 shows the selection & inclusion of item in the association rule set based on highest support count and deletion of rows where selected item is absent(value is 0). [3],[4]

Transaction	I2	I3	I4	I5
T1	1	1	0	1
~~T3~~	~~0~~	~~1~~	~~1~~	~~0~~
T4	1	0	0	1
T5	1	1	0	1
T6	0	0	0	1
T7	0	0	1	1
COUNT	3	3	2	5>=Min Sup

Rule Set = { I1, I5 }

ARB=0

Figure 8: Finding out second item to be selected in Rule Set from the updated transaction database

Figure 8 shows the selection & inclusion of 2nd item in the association rule set based on highest support count and deletion of rows where selected item is absent(value is 0)

Transaction	I2	I3	I4
T1	1	1	0
T4	1	0	0
T5	1	1	0
~~T6~~	~~0~~	~~0~~	~~0~~
~~T7~~	~~0~~	~~0~~	~~1~~
COUNT	3>=Min Sup	2	1

Rule Set = {I1, I2, I5}
ARB=0

Figure 9: Finding out third item to be selected in Rule Set from the updated transaction database

Figure 9 shows the selection & inclusion of 3rd item in the association rule set based on highest support count and deletion of rows where selected item is absent (value is 0).

Transaction	I3	I4
T1	1	0
T4	0	0
T5	1	0
COUNT	2 ≱ Min Sup	0

Figure 10: Reaching termination of the processing

Figure 10 shows the termination of the processing, as no item count is greater than or equals to Minimum Support=3(for this illustration)

Final Association Rule Set = {I1, I2, I5} (for this illustration)

Association rules	Confidence	Status
R1: I1^I2-> I5	Sc{I1,I2,I5}/Sc{I1^I2}=3/3=100%	Selected
R2: I2^I5->I1	Sc{I1,I2,I5}/Sc{I2,I5}=3/3=100%	Selected
R3: I1^I5->I2	Sc{I1,I2,I5}/Sc{I1,I5}=3/5=60%	Rejected
R4: I1->I2^I5	Sc{I1,I2,I5}/Sc{I1}=3/6=50%	Rejected
R5: I2->I1^I5	Sc{I1,I2,I5}/Sc{I2}=3/5=60%	Rejected
R6: I5->I1^I2	Sc{I1,I2,I5}/Sc{I5}=3/5=60%	Rejected

Figure 11: Binary transaction based final selection and rejection of association rules

Figure 11 shows final selection and rejection of association rules containing different combination of items comprising final Association Rule Set based on minimum confidence threshold 70% (for this illustration).

6. IMPLEMENTATION

Transaction database is considered at first for converting the item's presence in binary data. If any item ID is found within transaction, under the column of that item for the respective transaction the value 1 is dispersed, otherwise 0 is dispersed.
After the selection of any particular item as frequent item based on highest number of 1 present under that respective item column, the selected item is assigned in association rule set. After selection, the respective column values for that item is changed to a different level of value (some -99) along with the consequent rows for that column in which item is absent (Value 0), so that these rows & column won't be considered for next level of iteration along the processing following the same principle.
In the end, after reaching the termination of the processing, because of no item count is greater than or equals to Minimum Support, the final updated association rule set provides the item which can form different association rules. One more checking is required to analyze different association rules based on minimum confidence threshold.

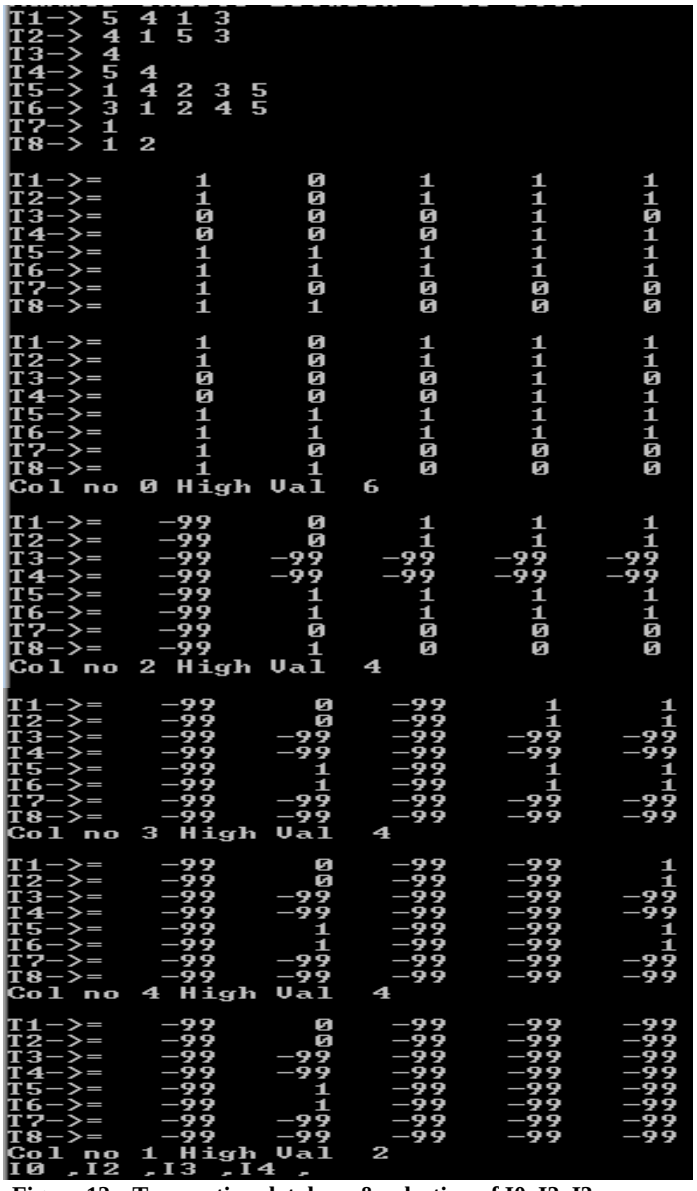

Figure 12:- Transaction database & selection of I0, I2, I3 & I4
sequentially based on highest number of 1's. (Minimum support threshold=3)

Figure 12 shows snapshot of one example transaction dataset, converting the dataset (signifying presence & absence of item in a particular transaction) into binary data & selection of different item one by one into association rule set.
"Col no" designates the item which is getting selected in the association rule set. "High Val" signifies the support count for that item.

7. CONCLUSION

In the Apriori algorithm one of the major shortcomings is that we have to access the database recurrently, which increases the running time of the algorithm.[2],[5] Keeping this thing in mind the modified algorithm have been developed to minimize the database scanning, which gives an accurate and efficient result. Although much work has been done in this regard earlier [6], but a new procedure have been presented which is simplistic yet efficient. After selection of an item that is being entered in the association rule set, particular transaction entry in which the selected item is absent is entirely rejected, which reduces the running time by a colossal factor, as these transactions won't be checked in subsequent stages of iteration. Because of this, it increases the processing speed which was the foremost target for developing a superior approach.

8. REFERENCES

[1] Han J, Kamber M, Pei J, Data Mining: Concepts and Techniques. Morgan Kaufmann Publisher.

[2] Sharma Sachin, Singhal Vidushi, Sharma Seema, A Systematic Approach and Algorithm for Frequent Data Itemsets, JGRCS, Vol 3, No. 11, Nov 2012.

[3] Kumar Vipin, Steinback Michael, Tan Pang-Ning, Introduction to Data Mining, Pearson.

[4] Sridevi R, Ramaraj E, Finding Frequent Patterns Based On Quantitative Binary Attributes Using FP-Growth Algorithm, IJERA, Vol.3, Issue 6, pp.829-834.

[5] Gupta G.K., Introduction to Data Mining with Case Studies, 2nd Edition, PHI

[6] Benelhadj Med El Hadi, Arour Khedija, Boufaida, Slimani Yahya, A binary based approach for generating association rules, Proceedings of the 2011 World Congress in Computer Science, Computer Engineering and Applied Computing.