Representation of Association Rule Mining of Apriori Algorithm using Graph Based Algorithm

Md. Arif Rahman, Tithi Bose, Mst. Farhana Rahman, Naima Enam Esha

Abstract— One of the most important problems in data mining is association rule mining. The Thesis aims at in depth study of Graph based approaches to find Association Rule Mining. The traditional Apriori algorithm uses an iterative technique. It scans the database several times. Each time it joins the k-1 itemset with itself and generates a candidate. Then prune the candidate to obtain k-itemset. It is very time killing and need more space for candidate. So primary goal of any Association Rule mining algorithm is to avoid candidate generation and reduce the database scanning as less as possible. To achieve this goal we design an algorithm which avoid any kind of candidate generation and scans the database only once. It uses an adjacency matrix to store information. At the time of scanning database the adjacency matrix is built simultaneously. We get all frequent itemset by only a single scan in the adjacency matrix. So the efficiency of the algorithm is better than traditional Apriori algorithm.

Index Terms— Data Mining, Association Rule Mining, Apriori Algorithm, Adjacency Matrix., Transaction, Itemset.

1 INTRODUCTION

Data mining can be viewed as a result of the natural evolution of information technology. This huge amount of information is stored in database systems. Sometimes the data needs to be used in particular sections. It becomes harder to extract the desired information from the large amount. Data mining is the process of extracting or mining knowledge from large amount of data. Data mining is mainly named as mining knowledge from data. Data mining has attracted a great deal of attention in the information industry and in society as a whole in recent years, due to the wide availability of huge amounts of data and the imminent need for turning such data into useful information and knowledge. The information and knowledge gained can be used for applications ranging from market analysis, statistics, fraud detection, science exploration, customer retention, production control, neural networks, database technology, machine learning, information retrieval etc. There are two types of data mining tasks: descriptive data mining tasks that describe the general properties of the existing data, and predictive data mining tasks that attempt to do predictions based on inference on available data.

Association rule mining (Aggarwal et.al. al [7]., 1993) is one of the important problems of data mining. The goal of the Association rule mining is to detect relationships or associations between specific values of categorical variables in large data sets. Association Rule is commonly used to find frequent itemset among a large amount of data. Frequent patterns are those which appear in a dataset frequently.

2 ASSOCIATION RULE

Association rule mining (Aggarwal et.al. al [7]. 1993) is one of the important problems of data mining. The goal of the Association rule mining is to detect relationships or associations between specific values of categorical variables in large data sets. This is a common task in many data mining projects. Suppose I is a set of items, D is a set of transactions, an association rule is an implication of the form X=>Y, where X, Y are subsets of I, and X, Y do not intersect. Each rule has two measures, support and confidence. Association rule mining was originally proposed in the domain of market basket data.

In general, Association Rule Mining can be viewed as a two step process as:

- Find all frequent Itemsets: By definition, each of these Itemsets will occur at least as frequently as a predetermined minimum support count, min sup.
- Generate strong association rules from the frequent Itemsets: By definition, these rules must satisfy minimum support and minimum confidence.

Basic concepts and Notation:

- Support: The support of an itemset is the fraction of the rows of the database that contain all of the items in the itemset. Support indicates the frequencies of the occurring patterns. Sometimes it is called frequency. Support is simply a probability that a randomly chosen transaction t contains both Itemsets A and B. Mathematically,

Support (A→B) = P (A ∩ B)

Confidence: Confidence denotes the strength of implication in the rule. Sometimes it is called accuracy. Confidence is simply a probability that an itemset B is purchased in a randomly chosen transaction t given that the itemset A is purchased. Mathematically,

Confidence (A→B) = P (B | A)

Generally, an association rules mining algorithm contains the following steps:

- The set of candidate k-Itemsets is generated by 1-extensions of the large (k -1) - Itemsets generated in the previous iteration.
- Supports for the candidate k-Itemsets are generated by a pass over the database.
- Itemsets that do not have the minimum support are discarded and the remaining Itemsets are called large k-Itemsets.
- This process is repeated until no more large Itemsets are found.

3 MARKET BASKET ANALYSIS

A typical example of frequent itemset mining is market basket analysis. This process analyzes customer buying habits by finding associations between the different items that customers place in their “shopping baskets” (in the form of X□Y, where X and Y are sets of items). The discovery of such associations can help retailers develop marketing strategies by gaining insight into which items are frequently purchased together by customers. So from the definitions,

- Association rule X→Y
- Support s = probability that a transaction contains X and Y
- Confidence c= conditional probability that a transaction having X also contains Y.

TABLE 1
LIST OF ITEMS IN A DATABASE

Transaction Id	Items
100	A, B, C
200	A, C
400	A, D
500	B, E, F

- A⇒C (s=50%, c=66.6%)
- C ⇒ A (s=50%, c=100%)

Apriori Algorithm is composed of two phases:

- Find all frequent Itemsets: By definition, each of these Itemsets will occur at least as frequently as a predetermined minimum support count.
- Generate strong association rules from the frequent Itemsets: By definition, these rules must satisfy minimum support and minimum confidence.

4 GRAPH BASED APPROACH TO FIND ASSOCIATION RULE

4.1 Primitive Association Rule:

Yen et. Al [11] has purposed a way to find out association rule mining using graph. The concept of graph and bit vector is used to find frequent itemset. A primitive association rule is an association rule which describes the association among database items which appear in the database. A primitive association pattern is a large itemset in which each item is a database item.

4.2 Primitive Association Graph Construction

•If an itemset is not a large itemset, then any itemset which contains this itemset cannot be a large itemset.

•For a large itemset (i1; i2; . . . ; ik), if there is no directed edge from item ik to an item v, then itemset (i1; i2; . . . ; ik; v) cannot be a large itemset.

Suppose (i1; i2; . . . ; ik) is a large k-itemset. If there is no directed edge from item ik to an item v, then the itemset need not be extended into k. 1-itemset, because .i1; i2; . . . ; ik; v. must not be a large itemset. If there is a directed edge from item ik to an item u, then the itemset (i1; i2; . . . ; ik) is extended into k+1-itemset (i1; i2; . . . ik; u). The itemset (i1; i2; . . . ; ik; u) is a large k+1-itemset if the number of 1s in BVi1 ^BVi2 ^ . . . ^ BVik ^ BVu achieves the minimum support. If no large k+1-itemsets can be generated, the algorithm terminates.

4.3 Examples

Consider the database shown in Table 2, where TID represents transactions and Items represents list of items against each transaction. the numbers of the items are 1, 2, 3, 4, 5. In the large item generation phase, the large items found in the database are items 1, 2, 3, 4, 5 and BV1, BV2, BV3,BV4 BV5 are 11110101,10111011,01101101,00010010, 10100000 respectively. For finding large two items use logical AND operation between large 1 Itemsets.

BV1^BV2= 11110101 ^ 10111011 =10110001(4)

BV1^BV3= 11110101 ^ 01101101=01100101 (4)

BV1^BV4= 11110101 ^00010010 =00010000 (1)

BV1^BV5= 11110101 ^10100000=10100000(2)

BV2 ^ BV3=10111011 ^ 01101101=00101001(3)

BV2 ^ BV4=10111011 ^00010010=00010000 (1)

BV2^ BV5=10111011^10100000=10100000(2)

BV3 ^ BV4= 01101101 ^00010010=00000000 (0)

BV3^BV5=01101101^10100000=00100000(1)

BV4^BV5=00010010^10100000=00000000(0)

TABLE 2
LIST OF ITEMS

TID	Items
T 100	1,2,5
T200	1,3
T300	1,2,3,5
T400	1,2,4
T500	2,3
T600	1,3
T700	2,4
T800	1,2,3

TABLE 3
BIT VECTOR FOR CORRESPONDING ITEMS

ITEMSETS (ITEM NO.)	BIT-VECTOR(BV)
1	11110101
2	10111011
3	01101101
4	00010010
5	10100000

So large two items are{ (1,2), (1,3), (1,5) (2,3), (2,5),}. And create an edge between these vertices, shown in fig 1.

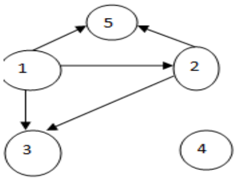


Figure 1: Association Graph of Table 3

Thus, the procedure is repeated until there is no frequent itemset in the database.

5 Proposed Graph based Algorithm For Association Rule Mining

5.1 Pre Requisites for Graph Based Approach

Graph: A linear graph G = (V,E) consists of a set of objects V = { v1,v2 } called vertices and another set E = { e1, e2 } whose elements are called edges, such that each edge is ek identified with an unordered pair (vi , vj) of vertices . The vertices vi, vj associated with edge ek are called the end vertices of ek.

Adjacency Matrix: The adjacency matrix of a graph G with n vertices and no parallel edges is an n by n symmetric binary matrix X = [xij] defined over the ring of integers such that

xij = 1 , if there is an edge between ith and jth vertices
= 0 , if there is no edge between them.

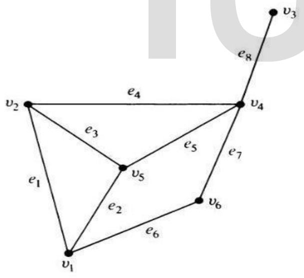


Figure 2: Simple Graph

$$X = \begin{array}{c|cccccc} & v_1 & v_2 & v_3 & v_4 & v_5 & v_6 \\ \hline v_1 & 0 & 1 & 0 & 0 & 1 & 1 \\ v_2 & 1 & 0 & 0 & 1 & 1 & 0 \\ v_3 & 0 & 0 & 0 & 1 & 0 & 0 \\ v_4 & 0 & 1 & 1 & 0 & 1 & 1 \\ v_5 & 1 & 1 & 0 & 1 & 0 & 0 \\ v_6 & 1 & 0 & 0 & 1 & 0 & 0 \end{array}$$

Figure 3: Adjacency Matrix of Graph of Figure 2

5.2 Proposed Algorithm

Algorithm One: Construction of Directed Graph

```
Input: Database D, Item n;
Output: An Adjacency Matrix Containing support count;
Begin:
Convert all item of database into string and insert them in hash table as key and set associated value in sorted order increasing by 1and start from 1.
For each transaction
   //n is the total item in the transaction
For i = 1 to n-2
For j = i+1 to n-1
      adj [ i , j ]++
x=merge(i,j)
For k = j+1 to n
       If (x is not in hash table as key)
 Place it in the hash table as key and increase the current associated value by 1;
m = associated value for key value x;
adj [ m , k]++
x = merge ( x, k) End
Procedure merge (item 1, item 2) will return a string "item 1-item 2.
```

5.3 Algorithm Two

```
Input: An initialized Adjacency Matrix

Output: Frequent k-itemset (k=2 to maximum possible item-set)

Begin:

for i = 1 to n // n is total item of database; for j = i+1 to max associated value of hash table if (adj [ i , j ]>=min_sup
        Result ∪{x,y}

//x=item get from key from hash table associated with i
//y= item get from key from hash table associated with j

End
```

5.4 Getting Frequent Itemset by Adjacency Matrix

While mining, the whole database, shown in Table 2 is converted into a directed graph, which is the adjacency matrix, adj. Adjacency matrix adj. can be expressed as adj [x, y] =c where c is the support count between x and y item. Every item against any transaction is assigned an integer number. And the integer number is fixed for the respective item in any transaction. So the frequent item set can be easily measured in data set. Integer numbers are sorted increasingly. Items are used as a node in the graph.

These values are used in adjacency matrix. Again these values are converted into string to get back the merged items.

We scan the adjacency matrix with the help of the second algorithm and take the items which satisfy the minimum support count. We get 2- itmeset to k-itemset (the maximum possible itemset). If Adj[x, y] satisfies minimum support and x contain m items we get (m+1) - itmeset otherwise 2- itemset. Then adjacency matrix is constructed as Table 4.

TABLE 4
ADJACENCY MATRIX OF THE DATABASE

Item Item	1	2	3	4	5	6	7	8	9	10
1	0	4	4	1	2	2	2	1	1	1
2	0	0	3	2	2	0	0	1	0	0
3	0	0	0	0	1	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0

Here total item is 5 and they are 1, 2,3,4,5.

Number of column = maximum associated value in the hash table.

In the first row 6, 7, 8,9,10 are not items but they are combination of items.

5= combination of item 2 & item 5.

6= combination of item 2 & item 3.

7= combination of item 3 & item 5.

8= combination of item 2, item 3 & item 5.

9= combination of item 2 & item 4.

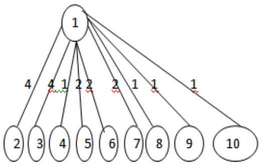


Figure: for row 1,(item 1)

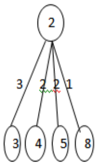


Figure: for row 3(item 3)

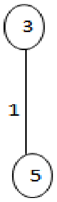


Figure: for row 3(item 3)

Figure: for row 4(item 4)

Figure: for row 5(item 5)

Figure 2: Figure of Adjacency Matrix (Table 4)

5.5 When new Transaction is Added to the Database

If any transaction or item is added to the database, our proposed algorithm does not need to scan the whole database once again. It just scans the newly added transactions or itemset and finds the frequent itemset among them. That is, it adds a new row in the Adjacency Matrix Table which is shown in Table 4. It then adapts the frequent itemset with the previous frequent itemset. So, it can be said that the proposed Algorithm scans the database for once.

5.6 Various Criteria of Transactions

Our proposed Algorithm satisfies the following four criteria:

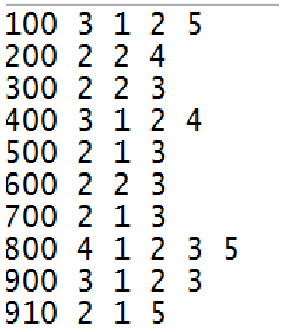
- Large Transaction, Large Itemset: when the transaction is large and also the number of items is Large.
- Small Transaction, Small Itemset: when the transaction is small and number of items is also small.
- Large Transaction, Small Itemset: when the transaction is large and number of items is small.
- Small Transaction, Large Itemset: when the transaction is small and number of items is large.

6 RESULT AND PERFORMANCE ANALYSIS

Here we implement 3(three) association rule mining algorithm in C++ including our proposed algorithm for finding frequent-2 itemset.

6.1 Result

We have taken a text file as a sample input for the entire three algorithms.



```
100 3 1 2 5
200 2 2 4
300 2 2 3
400 3 1 2 4
500 2 1 3
600 2 2 3
700 2 1 3
800 4 1 2 3 5
900 3 1 2 3
910 2 1 5
```

Figure 3: Sample Input

TABLE 5
SAMPLE OUTPUT

Description	Time Taken in Seconds
Apriori	0.15
Primitive	10.74
Adjacency Matrix	0.07

- Large Transaction, Large Itemset: Here, sample input has been taken as large transactions and Large Itemsets like- -60 Transactions and 49 Items.

Description	Time Taken in Seconds
Apriori	1.74
Primitive	11.484
Adjacency Matrix	1.697

- Small Transaction, Small Itemset: Here, sample input has been taken as small transactions and small Itemsets like- -05 Transactions and 05 Items.

Description	Time Taken in Seconds
Apriori	0.08
Primitive	10.304
Adjacency Matrix	0.07

- Large Transaction, Small Itemset: Here, sample input has been taken as small transactions and small Itemsets like- -50 Transactions and 05 Items.

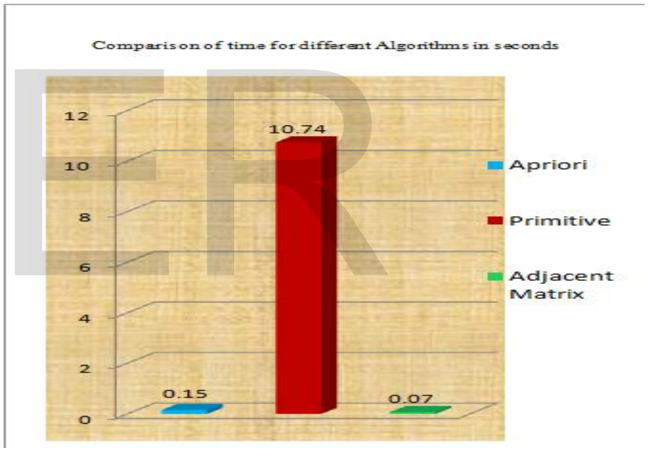
Description	Time Taken in Seconds
Apriori	0.11
Primitive	11.551
Adjacency Matrix	0.08

- Small Transaction, Large Itemset: Here, sample input has been taken as small transactions and small Itemsets like- -05 Transactions and 50 Items.

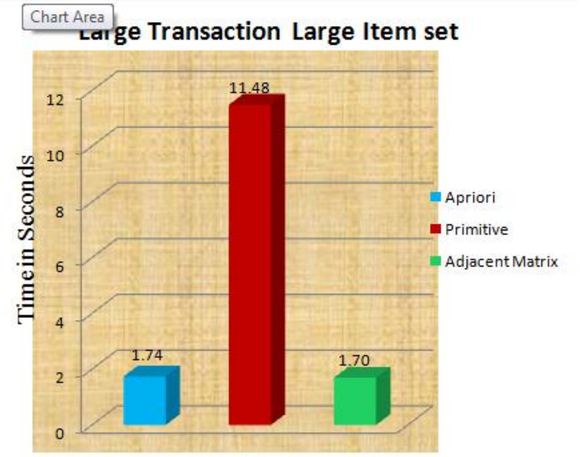
Description	Time Taken in Seconds
Apriori	1.31
Primitive	11.731
Adjacency Matrix	1.30

6.2 Performance Analysis

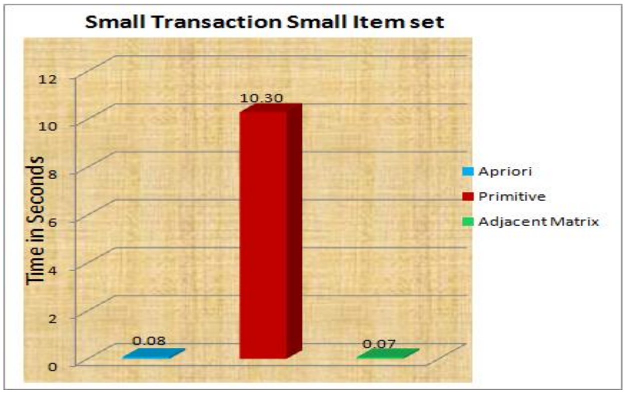
- For given Sample Input:



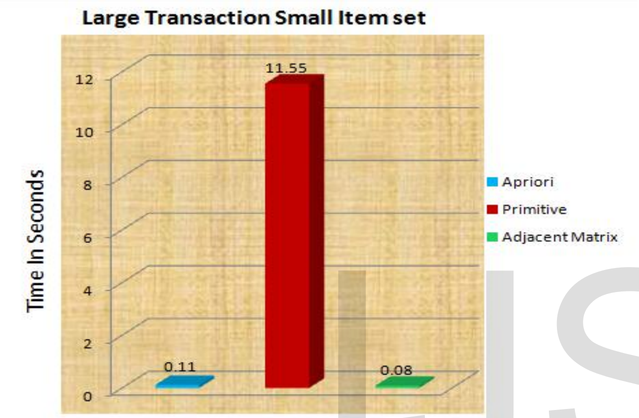
- Large Transaction, Large Itemset(60 Transactions, 49 Items)



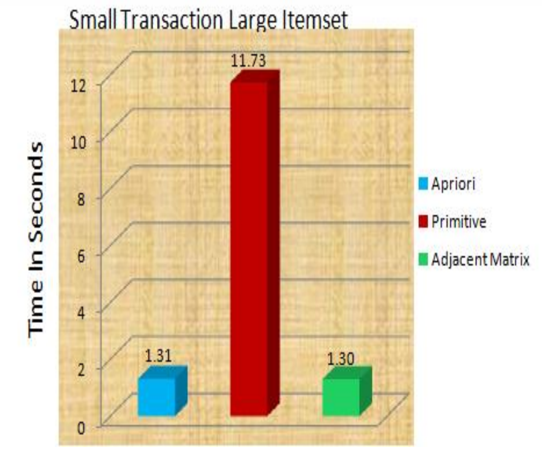
- Small Transaction, Small Itemset(5 Transactions, 5 Items)



- Large Transaction, Small Itemsets(50 Transactions, 05 Items)



- Small Transaction, Large Itemset(05 Transactions, 50 Items)



7 Conclusion

Association rule mining is the most efficient data mining tool. It should be implemented randomly such a way that almost all the requirements should be considered. Our proposal will be helpful to mine a large amount of data as well as small data as it scans the data for once.

References

[1] X. Wu, V. Kumar, J. Ross Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, P. S. Yu, Z.-H. Zhou, M. Steinbach, D. J. Hand, and D. Steinberg, "Top 10 algorithms in data mining," Knowledge and Information Systems, vol. 14, no. 1, pp. 1–37, (Dec. 2007).

[2] R. Srikant, "Fast algorithms for mining association rules and sequential patterns," UNIVERSITY OF WISCONSIN, (1996).

[3] J. Han, M. Kamber, "Data Mining: Concepts and Techniques", Morgan Kaufmann Publishers, Book, (2006).

[4] Rakesh Agrawal, Ramakrishnan Srikant, "Fast Algorithms for Mining Association Rules", Proceedings of the 20th VLDB conference, Santiago, pp 487-499, (1994).

[5] K.Sun and F.Bai,"Mining Weighted Association Rules Without Preassigned Weights", IEEE Transactions on Knowledge and Data Engineering, Vol. 20, No. 4, pages 489-495 , (April. 2008).

[6] Show-Jane Yen, Arbee L. P. Chen: "A Graph-Based Approach for Discovering Various Types of Association Rules". IEEE Trans. Knowl. Data Eng. 13(5): 839-845 (2001).

[7] Sotiris Kotsiantis, Dimitris Kanellopoulos, "Association Rules Mining: A Recent Overview", GESTS International conference on computer science and engineering, Vol.32 (1) pp- 71-82, (2006).

[8] Rakesh Aggarwal , Ramakrishanan Srikant, "Fast Algorithm for mining Association Rules", IBM Almaden Research Centre, Proceedings of 20th VLDB Conference, Santiago, Chile, (1994).

[9] Hemant Kumar Sharma, "GRAPH BASED APPROACHES USED IN ASSOCIATION RULE MINING".

[10] Anurag Choubey[1] , Dr. Ravindra Patel[2] and Dr. J.L. Rana[3] , "GRAPH BASED NEW APPROACH FOR FREQUENT PATTERN MINING", International Journal of Computer Science & Information Technology (IJCSIT) Vol.4, No 1, (Feb 2012).

[11] Show-Jane Yen, Arbee L. P. Chen: "A Graph-Based Approach for Discovering Various Types of Association Rules". IEEE Trans. Knowl. Data Eng. 13(5): 839-845 (2001).

[12] http://en.wikipedia.org/wiki/Association_rule_mining

Authors Information

- Md. Arif Rahman is an Assistant Professor in Computer Science and Engineering(CSE) dept. in Jessore University of Science and Technology(JUST) Jessore-7408 , Bangladesh. E-mail:ma.rahman@just.edu.bd. Cell:+880-01717-283829.Website:www.just.edu.bd

- Tithi Bose is currently pursuing masters degree program in Computer Science and Engineering in Jessore University of Science and Technology (JUST), Bangladesh. E-mail: tithibose2013@gmail.com

- Naima Enam Esha and Mst. Farhana Rahman both completed B.Sc. degree in Computer Science and Engineering in Jessore University of Science and Technology (JUST), Bangladesh. E-mail: naima_enam21@yahoo.com, farhana.rahman54@gmail.com.