A MATRIX BASED NEW ALGORITHM TO MINE ASSOCIATION RULE MINING IN LARGE DATABASES

Kanak Chandra Bora
Assistant professor,
Department of Computer Science & Electronics,
University of Science & Technology Meghalaya, 9th Mile, Ri-Bhoi, India

Abstract: ***Association rule mining has been occupying as one of the interesting field of research in data mining relating to the real life problem, market basket analysis. Various algorithms have been developed to find association rule in business transactions and each one having its own advantage and disadvantages. In this paper, a new matrix based algorithm has been developed to mine association rule in large databases which is more efficient in terms of database scanning and memory than existing one.***

Keywords: Association rule, frequent pattern, large itemset, market basket analysis, matrix based algorithm.

1: Introduction

Data mining means extracting hidden information from a large database which is required by top executive of an organization to take strategic decisions for the fulfilment of aim and objective of the organisation. . Association rule mining is an important area of data mining and it is used by retail store community. The purchasing of one product along with another product represents an association rule. Bar-code technology has made the retailers to collect and store massive amount of transactional data. Successful retail organisations use database technology to understand buying pattern of customers.

Mining of association rule is not only applied to market basket analysis. It has many applications in other areas like – in medical databases where information regarding the patients is stored can be applied association rule mining to understand what disease may come along with other diseases. For example suppose a person is suffering from sugar and hypertension, and then along with these two, another disease kidney problem may come. In addition to this, association rules are applied in diagnosing hyperlepidemia[1] and to understand what drugs are co-prescribed with antacid.

Weather forecasting means predicting weather for future. There are two types of predictions, descriptive as well as predictive. In predictive model of forecasting, prediction is performed based on historical data analysis. So in weather forecasting databases, association rule mining can be applied to detect the weather trend as a historical analysis. Again in bank loan databases, association rule mining can be applied to determine the characteristics of defaulters and based on what bank executive take decision regarding loan sanction.

Nowadays, terrorism is a big problem in our society. If big database is maintained regarding each family, applying association rule mining people can understand the different factors behind it and accordingly government takes initiative to remove those factors.

In the field of agriculture also if a database is maintained regarding different types of seeds, type and area of land, irrigation facility along with applied medicines, urea etc and productivity. Applying association rule mining one can understand what medicine how much quantity for which type of land for which seeds can produce maximum production.

In automobile repairing centre, if a database is maintained regarding different factors along with their various problems, then association rule can help in diagnosis vehicles fault.

It has been reported that association rule mining can be applied in other area of data mining. Even in classification problems, hidden knowledge discovery algorithm such as association rules mining algorithms can be applied successfully [2-4].

Mining association rules comprises two step processes. In the first step large itemsets are generated. In the second step association rules are generated from the large itemsets. To find large itemsets various algorithms have been developed which are used in finding association rules. The AIS was the first algorithm to generate association rule which was developed by R. Agrawal, T. Imielinski, and A. Swami in 1993[5]. The disadvantages of AIS algorithm are generating so many candidate item sets which are finally not in the frequent item sets due to small and the requirement of scanning the database many times. To overcome these disadvantages, later on the AIS algorithm was improved by R. Agrawal and R. Srikant[6] in 1994 and renamed the algorithm as Apriori. This algorithm requires repeated scanning of the input database over the entire frequent itemset mining process. A variation of Apriori algorithm was AprioriTid(transaction id). In AprioriTid transactions in the database are replaced by candidate itemsets that occur in that transaction. AprioriTid was proposed by R. Agrawal ans R, Srikant[7]. Apriori-hybrid was developed by R. Agrawal and R. Srikant by considering the advantageous features of Apriori and AprioriTid. M. Houtsma and A. Swami proposed another algorithm, the SEMT algorithm in 1993 to generate large itemsets using SQL [8]. Apriori algorithm was improved by many researchers[9]. A Direct Hash Based algorithm efficiently generates large itemsets reducing dataset size was introduced by Park, J. S. et al. [10]. In 1997, Soo et al developed an effective Direct Hashing and Pruning algorithm for mining association rules[11]. This algorithm progressively reduces the database size as well as avoids database scans in some passes. Another novel hash based approach algorithm for mining frequent itemsets over data stream was proposed by En et al[12].

A Matrix based algorithm presenting either 0 or 1 for absent or present respectively for items in a database for association rule mining in 2005 was introduced by Yuan, Y., Huang, T.[13]. This algorithm generates frequent candidate sets from which association rules are then mined. Again in 2007, L. Hanbing and W. Baisheng Wang developed an association rule mining algorithm using Boolean matrix[14], which is another variation of matrix based algorithm.

To overcome the drawback of Apriori series algorithms which require generation of candidate itemsets and scanning the data base many times, FP-Tree(Frequent Pattern Tree) algorithm was introduced by Han et. al in 2000[15]. This FP-Tree algorithm avoids the generation of

candidate itemsets and generates the FP-Tree scanning the database only two times. FP-Tree is mined to generate frequent itemsets. This algorithm is order of magnitude faster than the Apriori algorithm. The main disadvantages associated with FP-Tree algorithm are that the construction of FP-Tree is a time consuming process [16]. The process is not flexible and non repetitive as well as non reusable during the mining operation.

After generating large itemsets association rules are generated as per support and confidence [17].

Support: The support (s) for an association rule A => B is the percentage of transactions in the database that contains A U B.

Confidence: The confidence or strength (∞) for an association rule A => B is the ratio of the number of transactions that contain AUB to the number of transactions that contain A.

This paper is organised in three sections. Section-1 is the introductory part which covers applications of association rules mining in different areas along with different associations rule mining algorithms. New findings are discussed in section-2. A new algorithm is developed in section-3 and experimental result is also cited here including conclusion and future works.

2. NEW FINDINGS

The matrix based algorithm which was proposed by Hanbing, L., et al [14] cannot generate all the frequent itemsets in some databases. This is as per proposition 3, which is stated below in section-3. Applying this algorithm to the database which is considered in the experimental result will not be able to generate all large itemsets. One new matrix based algorithm is developed in section-3 along with three propositions.

3. **PROPOSED NEW ALGORITHM**

Here a Boolean matrix based new algorithm has been developed for finding large itemsets in databases.

BOOLEAN MATRIX: A Boolean matrix is defined as m × n matrix where m represents the number of transactions and n represents the number of items. Element of the matrix is either 1 or 0 based on whether a particular item is involved in a particular transaction or not.

Example: Suppose there is a database regarding transaction as follows

T100 a, b, c
T200 b, e
T300 a, d, e
T400 b, d

The Boolean matrix of the above database will be as shown in table-1.

table 1

	A	B	C	D	E
T100	1	1	1	0	0
T200	0	1	0	0	1
T300	1	0	0	1	0
T400	0	1	0	1	0

PROPOSITION 1: Support count of any item is less than the minimum support, then this item is removed from the Boolean matrix which will not affect the generation of large itemsets in the database.

Rationale: Minimum support is the threshold value. Items which have support count and is less than the minimum support cannot appear in the large itemsets. So unimportant items keeping in the Boolean matrix is avoided.
Hence the proposition.

PROPOSITION 2: Any row sum which is equal to 1 can be deleted from the Boolean matrix without affecting association rule generation.
Rationale: As association rule presents when one item comes along with another item in a transaction so single item in a transaction cannot take part in generation of association rule. Hence it can be deleted without any effect.

PROPOSITION 3: Proposition 1 & 2 are applied once in the beginning of generating large itemsets. Thereafter application of proposition 1 & 2 affect the production of large itemsets.
Rationale: In case proposition 1 & 2 are applied more than one then some items get deleted which would have participated in large itemsets. This will indirectly affect the generation of association rule.

3.1 Details of the Algorithm:

Step-1**:** Create the Boolean matrix for the given transactional database.
Step-2: Support of each item is calculated by counting the number of 1in each column in the Boolean matrix.
Step-3: column sum and row sum are calculated. As per proposition 1, any item which has row sum < minimum support are deleted from the Boolean matrix. As per proposition 2, any transaction whose row sum ≤ 1, are deleted from the Boolean matrix
Step-4: Items which are in the Boolean matrix are in the large itemset-1, L_1.
Step-5: Combination of 2, 3, ... up to the number of items presents in the Boolean matrix, are generated and their frequency are tested for minimum support. Itemsets having support count greater than equal to minimum support will be in the large itemsets L_2, L_3, ...
Step-6: Large itemset L = L_1U L_2 U L_3 .....

3.2 EXPERIMENTAL RESULT:

Example: A gent's clothing store one day transactions are recorded as follows.

table 2: Sample Clothing Transactions

Transactions	Items	Transactions	Items
T1	Genjee	T11	TShirt
T2	Shoes, LPent, TShirt	T12	Genjee, Jeans, Shoes, LPent, TShirt
T3	Jeans, TShirt	T13	Jeans, Shoes, HPent, TShirt
T4	Jeans, Shoes, TShirt	T14	Shoes, LPent, TShirt
T5	Jeans, HPent	T15	Jeans, TShirt
T6	Shoes, TShirt	T16	LPent, TShirt
T7	Jeans, LPent	T17	Genjee, Jeans, LPent
T8	Jeans, Shoes, HPent, TShirt	T18	Jeans, Shoes, HPent, TShirt
T9	Jeans	T19	Jeans
T10	Jeans, Shoes, TShirt	T20	Jeans, Shoes, HPent, TShirt

Suppose the minimum support s= 20% and confidence ∞ = 50 %
Now Boolean matrix is created for the above transactions table as shown in table-2 below.

TABLE 3

	Jeans	Ganjee	Shoes	HPent	LPent	TShirt	Row sum
T1	0	1	0	0	0	0	1
T2	0	0	1	0	1	1	3
T3	1	0	0	0	0	1	2
T4	1	0	1	0	0	1	3
T5	1	0	0	1	0	0	2
T6	0	0	1	0	0	1	2
T7	1	0	0	0	1	0	2
T8	1	0	1	1	0	1	4
T9	1	0	0	0	0	0	1
T10	1	0	1	0	0	1	3
T11	0	0	0	0	0	1	1
T12	1	1	1	0	1	1	5
T13	1	0	1	1	0	1	4
T14	0	0	1	0	1	1	3
T15	1	0	0	0	0	1	2
T16	0	0	0	0	1	1	2
T17	1	1	0	0	1	0	3
T18	1	0	1	1	0	1	4
T19	1	0	0	0	0	0	1
T20	1	0	1	1	0	1	4
Total	14	3	10	5	6	14	

As per step-2, total of each item is calculated at the last row. Total of each row is calculated at the last column. Now as per step-3, item Ganjee has support count 3 which is less than minimum support. So column attribute “Ganjee” will be deleted from the matrix and T1, T9, T11, T19 rows are deleted from the Boolean matrix. The new matrix will be as shown in table-3.

TABLE 4

	Jeans	Shoes	HPent	LPent	TShirt
T2	0	1	0	1	1
T3	1	0	0	0	1
T4	1	1	0	0	1
T5	1	0	1	0	0
T6	0	1	0	0	1
T7	1	0	0	1	0
T8	1	1	1	0	1
T10	1	1	0	0	1
T12	1	1	0	1	1
T13	1	1	1	0	1
T14	0	1	0	1	1
T15	1	0	0	0	1
T16	0	0	0	1	1
T17	1	0	0	1	0
T18	1	1	1	0	1
T20	1	1	1	0	1
Total	14	10	5	6	14

Now as per step-4, items which are present in the new Boolean matrix will be in L_1 as shown below.

L_1 = {Jeans}, {Shoes}, {HPent}, {LPent}, {TShirt}

Now as per step-5, combination of 2 items to 5 items one after another is tested for minimum support count and the combinations which have support count greater than the minimum support appear in the large itemsets as shown below.
L_2 = {Jeans, Shoes}, {Jeans, HPent}, {Jeans, TShirt}

L_3 = {Jeans, Shoes, Hpent}, {Jeans, Shoes, TShirt}

L_4 = {Jeans, Shoes, HPent, TShirt}
Again as per step-6,
L = {Jeans}, {Shoes}, {HPent}, {LPent}, {TShirt}, {Jeans, Shoes}, {Jeans, HPent}, {Jeans, TShirt}, (Jeans, Shoes, Hpent}, {Jeans, Shoes, TShirt}, {Jeans, Shoes, HPent, TShirt}
Large itemsets are generated and from these large itemsets association rules can be generated as per support and confidence.

3.3 Conclusion:

The drawback of the matrix based algorithm proposed by Yuan, Y., et al [13] was as follows.

This algorithm first generates large itemset-1 after generating candidate itemset-1. From the large itemset-1, candidate itemset-2 is generated and then large itemset-2 is generated and so on like Apriori algorithm. Here unnecessarily large numbers of candidate itemsets are generated. It is almost similar to Apriori instead of scanning the database; it will scan the matrix as many times as required.

In this new algorithm**, c**reating a Boolean matrix from the transactional database, large itemsets are generated without creating candidate itemsets. From these large itemsets, association rules are generated. Performance of this new algorithm is better than the existing Boolean based algorithm as it does not require generation of candidate itemsets. Again comparing with Apriori based algorithm it is observed that performance of this new algorithm is better as it does not require scanning the database many times as well as no requirement for generation of candidate itemsets.

3.4 Future Works:

Here only finding the large itemsets in a database is considered without considering quantity of items. Again as it is true that size of database going on increasing, hence parallelism is used to enhance the performance which is not considered here. So in future works these factors will be considered.

Acknowledgement:
The author would like to thank Dr. Bichitra Kalita for his valuable suggestion for writing research paper.

References:

1.Dogan, S., Turkoglu, I., "Diagnosing Hyperlipidemia using association rules", Mathematical and computational Applications, volume 13, No. 3, 2008, page 193-202.
2. Hu, Y.C., Chen, R.S., Tzeng, G.H., "Mining fuzzy associative rules for classifications problems", Computers and Industrial Engineering, 43 (4), 2002, page 735-750.
3. Li, J., Shen, H., Topor, R., "Mining the smallest association rule set for predictions", Proc. IEEE International Conference on Data Mining (ICDM 01), 2001.
4. Wang, Y., Wong, A.K.C.,"From Association to Classification: Inference Using Weight of Evidence", IEEE Transactions on Knowledge and Data Engineering, 15, 2003, page 764-767.
5. R. Agrawal, T. Imielinski, and A. Swami, " Mining Association Rules Between Sets of Items In Large Databases", In proceedings of the ACM SIGMOD International Conference on Management of data, page 207-216, 1993.
6. R. Agrawal, R. Srikant, "Fast algorithm for mining association rules in large databases". Proc. Of 20th Int'l conf. On VLDB: 487-499, 1994.
7. R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules in Large Databases", In Jorge B. Bocca, Matthias Jarke, and Carlo Zaniolo, editors, Proceeding of the 20th International Conference on Very Large Data Bases, VLDB, page 487-499, Santiago., Chile, Sptember 1994.
8. (SETM) M. Houtsma and A. Swami, "Set- oriented Mining of Association Rules in Relational Databases", IEEE International Conference on Data Engineering, page 25-33, 1993
9. Jiao Yabing, "Research of an Improved Apriori Algorithm in Data Mining Association Rules", International Journal of Computer and Communication Engineering, volume 2, No. 1, page 25-27, January 2013.
10. Park, J. S., Chen, M.S. and Yu P. S., " An Effective Hash Based Algorithm For Mining Association Rules", In Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data, M.J. Carey and D.A. Schneider, Eds. San Jose, California, page 175-186.
11. Soo J, Chen, M.S., and Yu P.S., "Using a Hash-Based Method with Transaction Trimming and Database Scan Reduction for Mining Association Rules", IEEE Transactions On Knowledge and Data Engineering, volume 5, 1997, page 813-825.
12. En Tzu Wang and Arbee L.P. ChenData, "A Novel Hash-Based Approach For Mining Frequent Item-Sets Over Data Streams Requring Less Memory Space", Data Mining and Knowledge Discovery, volume 19, number 1, page 132-172.
13. Yuan, Y., Huang, T.. A Matrix Algorithm for Mining Association Rules, Lecture Notes in Computer Science, volume 3664, September 2005, page 370-379.
14. L. Hanbing and W. Baisheng, "An Association rule mining Algorithm Based on a Boolean Matrix. "Data Science Journal, volume 6, Supplement, 9 September 2007, page 559-565.

15. Han, J., Pei, J., and Yin, Y., "Mining frequent patterns without candidate generation", in 2000 ACM SIGMOD Intl. Conference on Management of Data, W. Chen, J. Naughton, and P.A. Bernstein, Eds., ACM Press.
16. Saravanan Suba, Chistopher, T.,"A Study on Milestones of Association Rule Mining Algorithms in Large Databases",International Journal of Computer Applications, volume 47, No.3, June 2012, page 12-19
17. Dunham, Margaret H., "Data Mining – Introductory and Advanced Topics", PEARSON, Eleventh Impression, 2012.