Mining Association Rules Based on Boolean Algorithm - a Study in Large Databases

Chinta Someswara Rao, D. Ravi Babu, R. Shiva Shankar, V. Pradeep Kumar, J. Rajanikanth, and Ch. Chandra Sekhar

***Abstract*—With the wide applications of computers and automated data gathering tools, massive amounts of data have constantly collected in databases, which create immense demand for analyzing data and turning them into useful knowledge. Therefore, Knowledge Discovery and Data mining has become a research field in recent years to analyze the data in large databases. Association rule mining is one of the dominant methods for market basket analysis, which analyzes customer buying habits.**

The problem of association rule mining is that there are so many promising rules; it is obvious that such a vast amount of rules cannot be processed by inspecting each one. Therefore efficient algorithms restrict the search space and check only a subset of all rules. Boolean algorithm is one technique for mining association rules. The first objective of this study is to generate Association rules from massive databases in order to entrepreneurs can enlarge their own marketing strategies. The second objective of this study is to implement the program in the most efficient way in order to decrease the processing time and the memory consumption. This study can help retailers to build marketing strategies by gaining information about which items are frequently purchased together by customers.

***Index Terms*— KDD, databases, boolean algorithm.**

I. Introduction

Due to the rapid growth in the size and number of databases, there is a great need for discovering knowledge hidden in large databases, which create a new demand for examine such data and turning them into valuable perceptive. This perceptive spread widely into applications of computers like business management, administration, banking, social, health services, environmental protection and automated data collection tools [1]. Many kinds of knowledge can be mined from a database. Among them, the association rule is a very useful knowledge to be mined. The definition of an association rule is described in [2]

The Data mining supports large number of association rules. There is a difficulty to choose best one among available Association rules in data mining. It is difficult to process massive amounts of rules by inspecting each one. So to control the search space and to verify only a subset of all the rules, without missing important rules efficient algorithms are required [1]. Boolean Algorithm is one algorithm, shows very significant presentation over the other algorithms. In this study we framing the transactions with some max item size which are inserted into data base after that we have to analyze this transaction database by using Boolean algorithm.

Manuscript received January 18, 2013; revised May 28, 2013.

Chinta Someswara Rao, D. Ravi Babu, R. Shiva Shankar, V. Pradeep Kumar, J. Rajanikanth are with the SRKR Engineering College, Bhimavaram, AP, India (e-mail: chinta.someswararao@ gmail.com).

Ch. Chandra Sekhar was with AITAM, Tekkali, AP, India.

II. Related Work

Agrawal *et al*. proposed an algorithm, called AIS algorithm [3], for generating frequent itemsets. In the AIS algorithm, frequent itemsets are generated through iterations on scanning the database. The iteration terminates when no new frequent item-set is derived. After reading a transaction in the *k*th iteration, the AIS algorithm computes the candidate $k-$ itemsets by first deriving a set of $(k-1)-$itemsets which contains itemsets that are both in the frequent $(k-1)-$itemsets and in the transaction. One disadvantage of the AIS algorithm is that it generates too many invalid candidate itemsets. Houtsma and Swami proposed the SETM algorithm [4] that uses SQL for generating the frequent itemsets. Although it uses standard SQL join operation for generating candidate itemsets, the SETM algorithm generates candidate itemsets through a process of iterations similar to that of the AIS algorithm. The disadvantage of the SETM algorithm is similar to that of the AIS algorithm. That is, it generates too many invalid candidate itemsets.

In [3], the candidate item sets are generated on the fly during the pass over the database. For every transaction, candidate item sets are generated by extending the large item sets from previous pass with the items in the transaction such that the new item sets are contained in that transaction. In [2] candidate item sets are generated using only the large item sets from the previous pass. It is performed by joining the large item set with it. The resulting set is further pruned to exclude any item set whose subset is not contained in the previous large item sets. This technique produces a much smaller candidate set than the former technique. To count the supports for the candidate item sets, for each transaction the set of all candidate item sets that are contained in that transaction are identified. The counts for these item sets are then incremented by one.

Agrawal and Srikant also proposed two fast algorithms, called Apriori and AprioriTid [2], for generating frequent itemsets. In the Apriori algorithm, the candidate $k-$ itemsets is generated by a cross product of the frequent $(k-1)-$itemsets with itself. Then, the database is scanned for computing the count of the candidate $k-$ itemsets. The frequent $k-$ itemsets consist of only the candidate k $-$ itemsets with sufficient support. This process is repeated until no new candidate itemsets is generated. A hybrid algorithm is proposed which uses Apriori for initial passes and switches to

DOI: 10.7763/IJMLC.2013.V3.335

AprioriTid for later asses in [5]. In this paper we propose an object oriented approach for association rule mining in large databases.

III. Databases

Now-a-days many industries utilizing features of the data mining in large databases. One of the industries is retail market which is using data mining to analyzing the sales in the market [1]. The retail industries try to estimate the customer pulse with help of mine the data base which having buying trends of a customer. For example we have to consider one customer purchase some item like networks games, video games etc., over a period of time, by this to say that they will interest in the particular games area, so by this we have to estimate that they will purchase new games in the future. In this way say that data mining in data bases help the retail industry to identifying the recent trends [6]. At the same time data mining in databases helps industry to improve the sales as well as it also helps the industry to estimate the new customer pulse.

IV. System Architecture

As is shown in Fig. 1, data mining task can be achieved by applying Classification, Regression, Prediction, Clustering and Association rules. Association rule mining explore for interesting associations among the given items in a given data set [1]. In this study we have concentrated on Boolean Algorithm, which is an effective algorithm for mining Association rules in large transaction databases. In this study we have two phases for forming association rules:

The first phase: In this phase we provide number of transactions, max item set size and database name as input to the system. Based on this inputs this system scans the data base and follows number of transactions, maximum item set size it generates frequent item sets.

The second phase: In this phase we provide minimum support, minimum confidence and system automatically takes frequent item sets which are generated in first phase as input. Based in this data system Remove the rows of 1-item sets from frequent item sets which are not having possibility to be an antecedent.

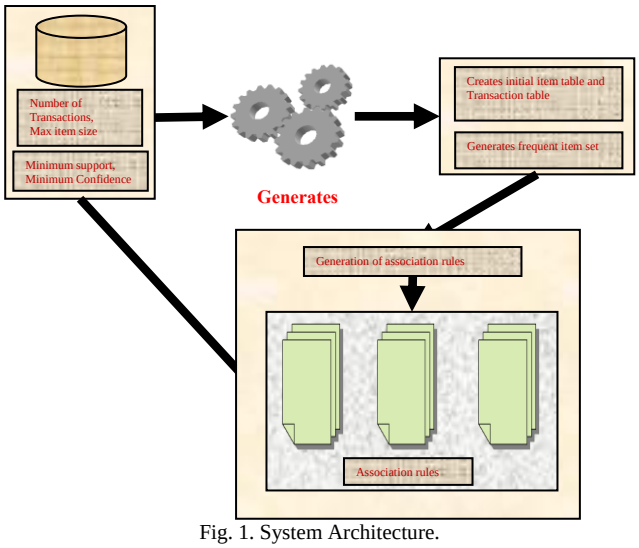

Fig. 1. System Architecture.

In this study we follow Boolean algorithm procedure. This Boolean algorithm easily counts the happening of 1s in each and every column of data. After counting the 1-item set the whole column will be eliminated whose corresponding column having only one value in the data. Applying logical and operation on 2 rows of data and the result is comparing with the one of the row which has least number of items. If they are equal, potential antecedent is taken as the frequent item set which is less number of items. If support of these two rows are greater than or equal to minimum confidence then association rules are generated. This procedure is repeated until there is no new rules are found.

This system is efficient, time saving and fast when compared to some of the other systems because it avoids generation of candidate item sets in each iteration i.e. generation of candidate item sets for each frequent k-item sets, which is burden for the processor if the database is huge. Moreover it requires only one scan of the database.

V. System Implementation and Results

Boolean algorithm supports two approaches.

A. Bit Stream Approach

Since each entry of the item table and transaction table can take only 0 or 1 as its values, we use one bit for an entry in both the item table and transaction table. Bit Stream approach drastically reduces the memory size and computation time for operations.

B. Sparse-Matrix Approach

This approach comparing the number of items in item set with small number of items in the other item sets. Also, an item set is usually related to a little number of transactions compared to the large number of transactions in the transaction set. As a result, an item table or a transaction table is very sparse. Using this approach, only non-zero entries in the tables need to be considered. It is also reducing the memory consumption and computation time.

C. Example

TABLE I: Transaction Data of the Transaction Database D

TID	Item sets
T1	A,D
T2	B,C,E
T3	A,B,C,E
T4	B,E
T5	A,B,C

	A	B	C	D	E
T1	1	0	0	1	0
T2	0	1	1	0	1
T3	1	1	1	0	1
T4	0	1	0	0	1
T5	1	1	1	0	0

Fig. 2. The boolean matrix A_{5*5}.

This section describes a sample execution of the proposed

algorithm. The transaction data of the transaction database D are given in Table I; the minimum support is 0.4; n=5 is the number of items, and m=5 is the number of transactions. Therefore, the minimum support number minsupsh=2. The transaction database D is transformed into the Boolean matrix A_{5*5} as shown in the Fig. 2.

We compute the sum of the element values of each column in the Boolean matrix A_{5*5} and the set of frequent 1-itemset is: L1= {{A}, {B}, {C}, {D}}

The fourth column of the Boolean matrix A_{5*5} is deleted because the support number of item D is smaller than the minimum support number 2. We then compute the sum of the element values of each row in the Boolean matrix and delete all rows where the sum of the element values is smaller than 2. Finally, the Boolean matrix A_{4*4} is generated as shown in Fig. 3.

	A	B	C	E
T2	0	1	1	1
T3	1	1	1	1
T4	0	1	0	1
T5	1	1	1	0

Fig. 3. The boolean matrix A_{4*4}.

The operation of 2-supports is executed for the all columns of the Boolean matrix A_{4*4}, and the set of frequent 2-itemset is: L2={{A,B},{A,C},{B,C},{B,E},{C,E}}

In pruning the Boolean matrix A_{4*4} by the set of frequent 2-itemsets L2, the third row of the Boolean matrix A_{4*4} is deleted because sum of its element values is smaller than 3. Finally, the Boolean matrix A_{3*4} is generated as shown in Fig. 4.

	A	B	C	E
T2	0	1	1	1
T3	1	1	1	1
T5	1	1	1	0

Fig. 4. The boolean matrix A_{3*4}.

The operation of 3-supports is executed for all columns of the Boolean matrix A_{3*4}, and the set of frequent 3-itemset is: L3= {{A, B, C}, {B, C, E}}

According to Proposition 3, the proposed algorithm is terminated because there are two frequent 3-itemsets in the set of frequent 3-itemset L3.

D. Implementation

This system is implemented in JAVA, specifically in java swings. It involves careful planning, investigation of the system and its constraints on implementation. Here we discussed some of the important classes which are involving to generate the mining association rules in large databases.

First window class: This class provides interface which is easily understand by the even normal person. Here we provide values like number of transactions, number of items and the database name as input. In this window we apply some limitations like numbers of transactions are in between 50 to 2000, max item size in between 3 to 10. First visit user must click on the generate data button to generate item sets. When user click on the Generate association rules button this class automatically calls the *synthaticdatabase* class with passing parameters like number of transactions, number of items and the database name.

Synthaticdatabase class: This class also provides the interface with accepting some input from previous class called *First window* and also from the user. Here we provide the minimum support and the minimum confidence as input. In this class also we apply some limitations like minimum support and minimum confidence in between 0 to1. When we press start button it automatically calls the *frequentitemset* class by passing parameters like minimum support and minimum confidence.

Frequentitemset classs: This interface automatically take the some of the inputs like number of transactions, number of items , the database name ,minimum support and minimum confidence. Based on this inputs it generate the initial transaction table as well as the 1-frequent item set, and also displays these tables on the interface along with the count vector , and the result is pass on to the next calling class.

K_itemsets class: This class also takes the some of the inputs like 1-frequent item set, transaction table etc., by calling the *frequentitemset* class. Based on this inputs it generates k-frequent item sets, and the result is pass on to the next calling class.

Associationrules class: This class is the most important in this implementation. It accepts the data from previous classes like *k_item set, frequentitemset* class etc., as input. Based on this data it generates the Final result called Association rules in the following manner. Eliminating the rows of single item sets from the given input data sets which have no opportunity to be subsequent of any Association rules. In this implementation we simply counts occurrence of 1's in each column of the data. In the next step we have to apply the logical and operation on available two rows say some A and C then compare the result with the one another of these two which has less no of items. If these two rows are equal, then we have to consider that less number of items set is a frequent item set. In the next step we calculate support of each row of A and C. After calculating we check condition if support of these two is greater than or equal to minimum confidence then based on this association rule is generated.

VI. Results

In this study we provide number of transactions, max number of items, database name, minimum support, and minimum confidence as input to our system. Based on this data system generates the association rules. In this study we conduct several experiments by changing different parameters. We observe results which are shown in Table II.

By observing the Table II we say that if number of transactions and item size increases execution time gradually increases. The second observation from the Table II is if numbers of transactions increase association rules are also increases. The third observation is that if the transactions are constant but the size of the item set vary, then the association rules does not change within a constant period of minimum support and minimum confidence. In this way we say that performance of Boolean algorithm is somehow better to forming the association rules in terms of execution time for different minimum supports as predefined threshold supports.

TABLE II: RESULTS

INPUT				OUTPUT	
Number of Transactions	**Maximum item size**	**Minimum support**	**Minimum confidence**	**Generated association rules (formed)**	**Execution time(sec)**
100	3 4 5 6 7	0.1	0.2	32 162 620 2130 6932	1 1 1 1 2
200	3 4 5 6 7	0.1	0.2	32 162 620 2130 6932	2 2 2 2 3
400	3 4 5 6 7	0.1	0.2	32 162 620 2130 6932	5 5 5 6 7
600	3 4 5 6 7	0.1	0.2	32 162 620 2130 6932	11 12 13 13 16
800	3 4 5 6 7	0.1	0.2	32 162 620 2130 6932	23 25 27 29 33
1000	3 4 5 6 7	0.1	0.2	32 162 620 2130 6932	46 47 49 51 62

In order to appraise the performance of the Boolean algorithm, we conducted an experiment using the Apriori algorithm and the Boolean algorithm. The algorithms were implemented in Java and tested on a Windows Professional platform. Fig. 5 presents the experimental results for different numbers of minimum supports. The results show that the performance of the Boolean algorithm is much better than that of the Apriori algorithm. Moreover, the better the performance efficiency of proposed algorithm is, the smaller the minimum support is. This is because the smaller the minimum support, the more candidate itemsets the Apriori algorithm has to determine, and also the Apriori algorithm's join and pruning processes take more time to execute. However, the Boolean algorithm does not produce candidate item sets, and it spends less time calculating k-supports with the Boolean matrix pruned.

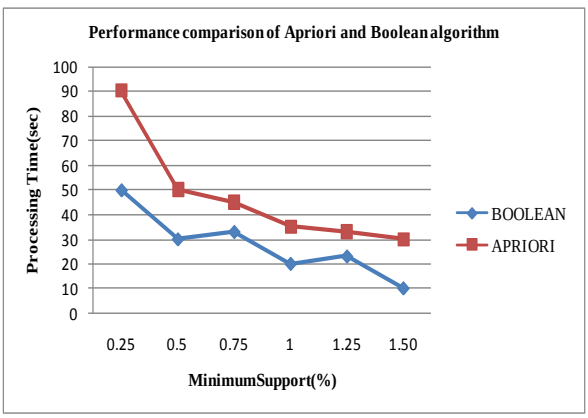

Fig. 5. Performance comparison of Apriori and Boolean algorithm.

VII. CONCLUSIONS

In this study we concentrate on mining association rules based on Boolean algorithm in large databases. One of the advantages of this study is to generate the frequent sets without generating the candidate item sets from a given data. We provide the easy and friendly interface to the persons. We also provide capable implementations of mining association rules based on Boolean algorithm in large databases. In this study we conduct number of experiments whose results shown that this study shows somehow better performance when compared to the other approaches at the same time this is also useful to retail market.

REFERENCES

[1] C. Sheng, Y. Jia, and C. Yang, "The research of improved apriori algorithm for mining association rules," in *Proc. International Conf. on Service Systems and Service Management*, 2007, pp. 1-4.

[2] R. Agrawal and R. Srikant, "Fast algorithms for mining Association rules in large databases," in *Proc. the 20th International Conf. on Very Large Data Bases*, 1994, pp. 487-499.

[3] R. Agrawal, T. Imielinski, and A. Swami, "Mining association rules between sets of items in large databases," in *Proc. International Conf. on Management of Data*, pp. 207–216, 1993.

[4] M. Houtsma and A. Swami., "Set-oriented mining for association rules in relational databases," in *Proc. IEEE 11th International Conf. on Data Engineering*, 1995, pp. 25–33.

[5] A. Savasere, E. Omiecinski, and S. Navathe, "An Efficcient Algorithm for Mining Association Rules in Large Databases," in *Proc. the 21th International Conf. on Very Large Data Bases*, 1995, pp. 432-444.

[6] C. C. Aggarwal, C. Procopiuc, and P. S. Yu, "Finding Localized Association in Market Basket Data," *IEEE Trans. on Knowledge and Data Engineering*, pp. 51-62, 2002.

Ch. Someswara Rao is currently an assistant professor in the department of computer science and engineering at the SRKR Engineering College, India. He received his master of science and master of technology degree from the Andhra University, Visakhapatnam, India. His M. Tech degree includes Computer Science and Technology. He is the co-principal investigator of a research project on "Implementation of Collaborative Product Development System Using Step/Xml" sponsored by Department of Science & Technology, Government of India, NEW DELHI. He is also the co-principal investigator of a research project on "XML ontology tool for STEP-NC/XML" sponsored by UGC, Government of India, NEW DELHI. His present researches include data mining, string matching, semantic web, XML-ontology, CAD/CAM, and their application to data exchange from STEP.

R. Shiva Shankar is currently an assistant professor in the department of computer science and engineering at the SRKR Engineering College (India). He received his master of technology degree from the Andhra University, Visakhapatnam, India. His M. Tech degree includes Computer Science and Technology.