An Innovative Approach to Association Rule Mining Using Graph and Clusters

N. Balaji Raja[1,*] and G. Balakrishnan[2]

[1] Department of Computer Applications,
J.J. College of Engineering & Technology, Tamil Nadu, India
nbalajiraja@yahoo.co.in
[2] Department of Computer Science and Engineering,
Indra Ganesan College of Engineering Tamil Nadu, India
director@igceng.org

Abstract. There is a common belief that the association rule mining has a major stake in data mining research domain. Numerous algorithms are proposed for discovering frequent itemsets. This is the key process in association rule mining. This proposed paper introduces two concepts. First leads to analyze the ability to explore the demographic parameters of the underlying entities and their inter relations using the traditional graph theory approach(vertices and edges). Second is the improved algorithm based on graph and clustering based mining association rules. This improved algorithm is named Graph and Clustering Based Association Rule Mining (GCBARM). The GCBARM algorithm scans the database of transaction only once to generate a cluster table and then clusters the transactions into cluster according to their length. The GCBARM algorithm is used to find frequent itemsets and will be extracted directly by scanning the cluster table. This method reduces memory requirement and time to retrieve the datasets and hence it is scalable for any large size of the database.

Keywords: Association rule mining, Data Mining, Relational, cluster, graph, GCBARM.

1 Introduction

With the rapid development in size and number of available databases in commercial, organizational, administrative and other applications [1, 19], it is the urgent need for new technologies and automated tools to change this wealth of data resources into useful information. The most challenging in database mining is developing fast and efficient algorithms that can deal with large volume of data because several data mining algorithm computation is used to solve diverse data mining problem generally know as associations, clustering, classifications and sequential patterns [2].

* Research Scholar, Anna University, Tamil Nadu, India.

N. Meghanathan et al. (Eds.): CCSIT 2012, Part III, LNICST 86, pp. 59–67, 2012.

1.1 Association Rule

Association rule mining is a very important research topic in the data mining field, it's problem in large databases is to generate all association rules [17, 20], of the form X =>Y, that will produce strong association rules which satisfy both minimum support degree (min_sup) and the minimum confidence degree (min_conf) greater then the user defined minimum support and minimum confidence [2, 3.4].

Definition 1. let X={x1, x2 ….. xn} be a set of items, then $D = \{< T_{id}, T > | T \subseteq X\}$ is a transaction database, where T_{id} is an identifier which be associated with each transaction.

Definition 2. Let $A \subseteq X, B \subseteq X, and A \cap B = \phi$, we call this $A \Rightarrow B$ as association rule.

Most of the algorithm generally is executed in two steps. First, to finding all sets of items that have support above the given minimum, and then generating the desired rules from these item sets. The apriori algorithm is the same as the association rule. The first step is to find all frequent itemsets, the next is to generate strong association rules from frequent item sets during pruning [18].

1.2 Apriori Algorithm

The apriori algorithm is a fast algorithm for mining association rules and is based on [5,20] algorithms for mining association rules, the problem of this algorithm is number of data scans n, where n is the size of large nonempty itemset and number of discovering rules is huge while most of the rules are not interesting. Therefore, after apriori algorithm many improved efficient versions were proposed towards scalability.

This paper introduces an algorithm GCBARM, which is basically different from previous algorithms; the remaining paper is organized as follows: related work is presented in Section 2, Section 3 gives the details of GCBARM algorithm with example, Section 4 show the experimental result and finally the conclusion is given in Section 5.

2 Literature Review on Association Rule and Graph database

Existing studies in data mining have presented efficient algorithm for discovering association rules. But the main drawback of the first algorithm is the need to do multiple passes over the datasets to generate frequent itemsets. The apriori association rule algorithm proposed by Agrawal and Srikant [6] can discover meaningful itemsets, but a large number of the candidate itemsets are generated from single itemsets and level by level in the process of creating association rules. Performance is severely affected because the database is scanned repeatedly to each candidate itemset with the database.

FP – Growth [7] out performs all candidate set generation and test algorithms as it mines frequent patterns without candidate generation. The main problem is no common prefixes within the data items.

Sample algorithm reduces the scanning to the datasets, it's scans single scan, but wastes considerable time on candidate itemsets [8].

The column wise apriori algorithm [9] and the tree based association rule algorithm [10] transformed the storage structure of the data, to reduce the time of scans of the transaction database.

The partition algorithm to improve efficiency and reduce the database scans, but still wasted scans on infrequent candidate itemset [11].

The primitive association rule mining is mining that describe the association among items of the transaction in the database. A uniform frame work was framed to perform association rule of association rules[12].

In the generalized association patterns, one can add all ancestors for each items from concept hierarchy and then apply the algorithm on the extended transactions [13].

The tones of day to day customer transactions stored in a very large database are processed and discovered multiple level of transactions with relevant attributes. Each of these attribute represented certain concept. Combining these attributes later generates multiple level concepts [5].

For frequent patterns in this new graph model which we call taxonomy superimposed graphs, there may be many patterns that are implied by the generalization and specialization hierarchy of the associated node label taxonomy [14]

Graph databases are able to represent as graphs of any kind of information, where naturally accommodated changes in data can be possible [15, 16].

The disadvantage of these algorithms is

- Number of reads the database transaction n time data scans where n is the size of large nonempty itemset,
- It is an incompetent as it requires wastage memory.
- Huge non interesting rules are discovered..

The proposal method GCBARM is a consequence to overcome the above said drawbacks

3 Proposed Methodology

This is a chance to establish that cluster based algorithms are still available by providing a better graph data structure which is used to simplify the process of generating frequent k itemsets, where k>=2. The proposed Graph and Clustering Based Association Rule Mining (GCBARM) algorithm scans the database of transaction only once, which overcome the drawbacks of the previous algorithms.

The process of building the graph is given in sequential numbers, this simplified pre-process is taken in to consideration as an important action before applying our proposed algorithm. The GCBARM algorithm scans the database of transaction only once to generate a cluster table as a two dimensional array where the rows represent transactions' TIDs and the columns represent items. The presence and absence of an item in a transaction indicates that 1 and 0 is content of the table. Here, considering an example, as to predict task whether a person makes ever 50K a year, that person's personal data attributes such as age, work class, fnlwgt, education, educationnum, maritalstatus, occupation, relationship, race, sex, capitalgain, capitalloss, hoursperweek and nativecountry are mainly taken for analysis.

Initially an original database consists of three relations, but it's pre-processed and shown in table 1. Then the three relations are converted into equivalent graph databasse (GDB) shown in figure 1, using this procrssing to identify the attributes for rule mining. The database consists of three relations, named Adult Personnel Detail, Adult Employment Detail and Qualification. The data structure of the relations is

Adult Personnel Detail (Fnlwgt, Age, marital-status, relationship, race, sex, native-country)

Adult Employment Detail (Work class, fnlwgt, occupation, capital-gain, capital-loss, hours per week, prediction task 50k)

Qualification (Fnlwgt, Education, education num)

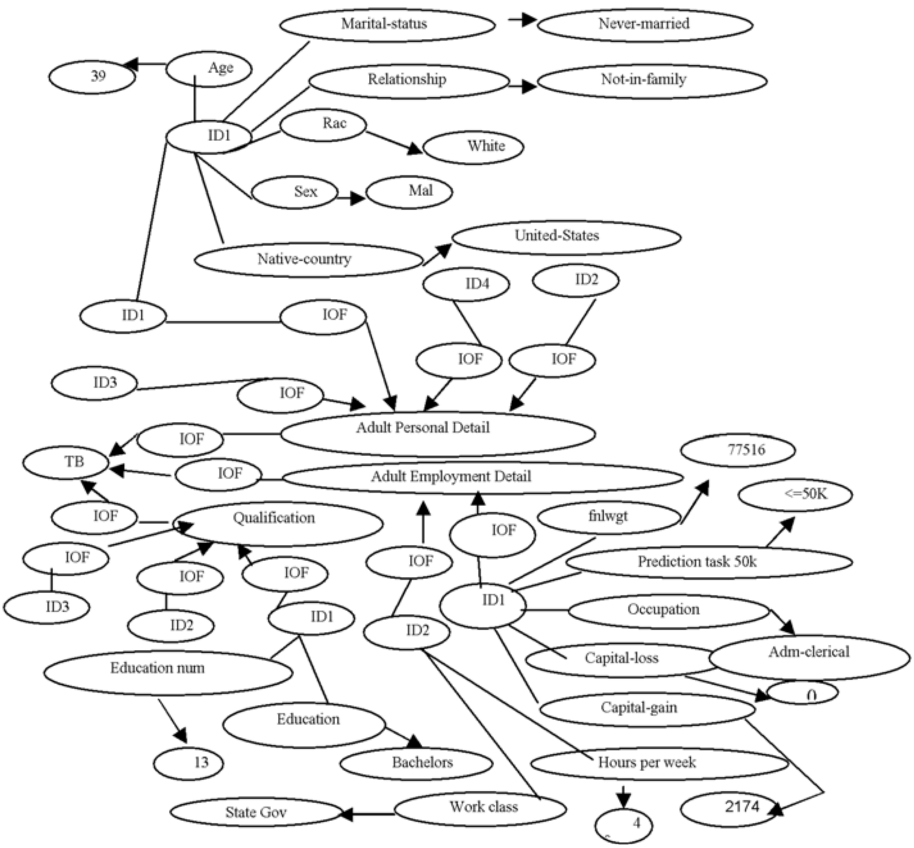

Fig. 1. Mapping of above relation into graph database

Table 1. An example of transaction database

TID	Age	work class	marital-status	native-country	P_task
T1	39	State-gov	Never-married	United-States	<=50K
T2	50	Self-emp-not-inc	Married-civ-spouse	United-States	<=50K
T3	38	Private	Divorced	United-States	<=50K
T4	53	Private	Married-civ-spouse	United-States	<=50K
T5	28	Private	Married-civ-spouse	Cuba	<=50K
T6	37	Private	Married-civ-spouse	United-States	<=50K
T7	49	Private	Married-spouse-absent	Jamaica	<=50K
T8	52	Self-emp-not-inc	Married-civ-spouse	United-States	>50K
T9	31	Private	Never-married	United-States	>50K
T10	42	Private	Married-civ-spouse	United-States	>50K
T11	37	Private	Married-civ-spouse	United-States	>50K
T12	30	State-gov	Married-civ-spouse	India	>50K
T13	23	Private	Never-married	United-States	<=50K
T14	54	Private	Separated	United-States	<=50K
T15	35	Federal-gov	Married-civ-spouse	United-States	<=50K
T16	43	Private	Married-civ-spouse	United-States	<=50K
T17	59	Private	Divorced	United-States	<=50K
T18	56	Local-gov	Married-civ-spouse	United-States	<=50K
T19	19	Private	Never-married	United-States	>50K
T20	23	Local-gov	Never-married	United-States	<=50K

After pre-processing the set minimum support threshold is 50%. There are 20 transactions and five different items named Age, Work class, Marital status, Native country and prediction task in the database. Most of the rule mining algorithms are in lexicographical order. An example of transaction database is shown in Table1.The items name of the transaction database is used rather than numbers to deal with some worst cases. The requirement of numbering system is to first scan the database to identify the length of each transaction, that means length of the numbers to the items in a transaction, and at the same time assigning the numbers to the items: Number 1 is assigned as item Age, Number 2 is assigned as item work class, Number 3 is assigned as marital status, Number 4 is assigned as native country, Number 5 is assigned as prediction task. This conversion process help us in both constructing the cluster table and building the graph, this process help avoid the need to rescan the transaction database. Next move is the clustering table that can easily reside in the main memory. In this example, the maximum transaction length is five, there will be at most five clusters, the total number of clusters is five as shown in table 2, The presence and absence of an item in a transaction is denoted by 1 and 0 in content of the table. After that , the bit vector for each item will be ready and it is an easy process to determine the frequent 1 itemsets by counting the number of 1s in each transaction, the minimum support threshold is not less than counting the number of 1s, but it is considered as a frequent itemset and then building the graph.

Table 2. The cluster table form the database in table 1

Item No. / TID	T7	T8	T9	T10	T14	T17	T19	T2	T3	T4	T5	T11	T12	T13	T16	T18	T1	T6	T20	T15
1	0	0	1	0	0	0	1	0	1	0	1	1	1	1	0	0	1	1	1	1
2	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	1	1
3	0	1	0	1	0	0	0	1	0	1	1	1	1	0	1	1	0	1	0	1
4	0	1	1	1	1	1	1	1	1	1	0	1	0	1	1	1	1	1	1	1
5	1	0	0	0	1	1	0	1	1	1	1	0	0	1	1	0	1	1	1	1

The bit vectors for the items are:

BV1=00100010101111001111
BV2=00000000000010011011
BV3=01010001011110110101
BV4=01111111110101111111
BV5=10001101111001101111

By counting the number of 1s in each bit vector is determined the support for each candidate itemset of length 1, as follows: support ({1}) = 55%, support ({2}) = 25%, support ({3}) = 55%, support ({4.}) = 85%, support ({5}) = 65%. Thus the frequent 1 itemsets are :{{1}, {3}, {4}, {5}} as their supports are not less than 50%.

The second step starts by reordering frequent 1 itemsets by providing each one with a sequential number to facilitate the process of constructing the graph, which is making logical( ) and operation between each pair of consecutive frequent 1 itemsets<$item_i$,$item_j$>l i<jif the number of 1s in the result is greater than or equal to minimum support threshold, a edge is directed to drawn from $item_i$ to $item_j$, this process is repeated for all frequent 1 itemsets. The simple directed graph to display frequent k – itemsets,k>=2 is shown in figure 2, and by assigning 25% as a new value to the minimum support threshold, the frequent 2 itemsets will be: {{1,3}, {1,4}, {3,5}}as shown in table 3 and the graph is constructed by drawing an edge between each pair of frequent items, as shown in figure 2. By counting the number of 1s in each bit vector is determined the support for each candidate itemset of length 2, as follows: support ({1,3}) = 25%, support ({1,4}) = 45%, support ({3,5}) = 30%. Thus the frequent 2 itemsets are :{{1,3}, {1,4}, {3,5}} as their supports are equal and above 25%.

The traverse of graph as if their path is to determine frequent 3 itemsets among three nodes{i,j} and {j,k} then the set {i,j,k} will be frequent 3 itemsets. Here, in this example, {{1,3,5}} is the only frequent 3 itemsets. As there are no extra edges, by assigning 12.5% as a new value to the minimum support threshold, the frequent 3 itemsets will be: {1,3,5}as shown in table 4 and the support for each candidate itemset of length 3, as follows: support ({1,3,5}) = 15%. Thus the frequent 3 itemsets are :{1,3,5} as their supports are equal and above 25%. Finally the algorithm terminates

Table 3. frequent itemsets 2 from frequent itemsets 1

Item No. / TID	T7	T8	T10	T14	T17	T18	T9	T19	T2	T3	T4	T12	T13	T16	T1	T20	T5	T11	T6	T15
{1,3}	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	1	1
{1,4}	0	0	0	0	0	0	1	1	0	1	0	0	1	0	1	1	0	1	1	1
{3,5}	0	0	0	0	0	0	0	0	1	0	1	0	0	1	0	0	1	0	1	1

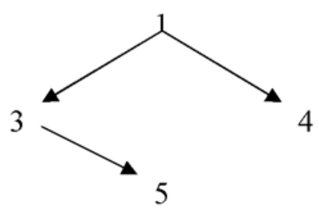

Fig. 2. A simple directed graph to display frequent k-itemsets, k>=2

Table 4. frequent itemsets 3 from frequent itemsets 1

Item No. / TID	T7	T8	T9	T10	T14	T17	T19	T2	T3	T4	T11	T12	T13	T16	T18	T1	T20	T5	T6	T15
{1,3,5}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

In this regard, as the database contains hundreds and thousands of transactions database and different items, constructing only one graph is not suitable for this work: So construct different graphs for each cluster and find from this graph all frequent itemsets, then combine the subsets of frequent itemsets together to get the whole set of frequent itemsets, and this technique is scalable with all transactions databases of different sizes.

4 Experimental Results

In order to appraise the performance of the proposed technique, which was implemented the GCBARM algorithm, along with Apriori algorithm using java programming language. The test databases are collected from UCI standard datasets available to evaluate rule mining algorithms.

Apriori and GCBARM are execute both algorithm at various values of minimum support thresholds, Figure 3 shows the average execution time (seconds) to generate all frequent itemsets using GCBARM and Apriori. The experimental results in figure 3 show the better performance of GCBARM algorithm than Apriori in terms of execution time.

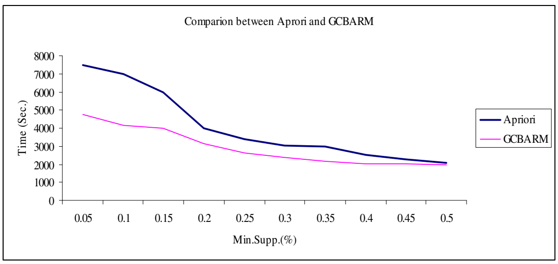

Fig. 3. Comparison between Apriori and GCBARM

5 Conclusion

This paper proposes the use of the graph database for pre-processing, after the hole transaction database is divided into partitions of variable sizes. Each cluster is considered one at a time by loading the first cluster into memory and calculating large itemsets and the corresponding support counts. Then the second cluster is loaded additionally and cumulative support count is then derived for the second clustered large itemsets. This process is continued for the entire set of clusters and finally the whole large itemsets and the corresponding cumulative support counts. This approach reduces main memory requirement since it considers only a small cluster at a time and hence it is scalable for any large size of the database.

References

1. Klosgen, W., Zytkow, J.M.: Hand Book of Data Mining and Knowledge Discovery. Oxford University Press (2002)
2. Kamber, M., Han, J.: Data Mining Concepts and Techniques. Morgan Kauffman Publishers, San Francisco (2001)
3. Imielinski, T., Agrawal, R., Swami, A.N.: Data Mining a Performance Perspective. IEEE Transactions on Knowledge and Data Engineering 5, 914–925 (1993)

4. Agrawal., R., Srikant, R.: Fast Algorithms for Mining Association Rules in Large Databases. In: Proceedings of the Twentieth International Conference on Very Large Databases, Santiago, Chile, pp. 487–499 (1994)
5. Han, J., Fu, Y.: Mining Multiple Level Association Rules in Large Databases. In: Proceeding of 1995 Int'l Conf. on Very Large Data Bases (VLDB 1995), Zurich, Switzerland, pp. 420–431 (1995)
6. Agrawal, R., Imilienski, T., Swami, A.: Mining association rules between sets of items in large databases. In: Proceedings of the ACM SIGMOD Int'l Conf. on Mgt of Data, Washington, DC, pp. 207–216 (1993)
7. Han, J., Pei, J., Yin, Y.: Mining frequent Patterns without Candidate Generation. In: ACM-SIGMOD, Dallas (2000)
8. Berzal, F., Cubero, J.C., Marin, N., Serrano, J.M.: TBAR: An Efficient Method for Association Rule Mining in Relational Databases. In: Elserier Data and Knowledge, Engineering, pp. 47–64 (2001)
9. Cheung, D.W., Han, J., Ng, V.T., Fu, A.W., Fu, Y.: A fast distributed algorithm for mining association rules. In: Proceedings of Int'l Conf. on PDIS 1996, Miami Beach, Florida, USA (1996)
10. Agrawal, R., Srikant, R.: Mining sequential patterns. In: Proceedings of the 11th Int'l Conf. on Data Engineering, ICDE (1995)
11. Franklin, B.: Genealogical Data Mining (2006)
12. Yen, S.-J., Chen, A.L.P.: A Graph based Approach for Discovering Various Types of Association Rules. IEEE Transaction on Knowledge and Data Engineering 13(5), 839–845 (2001)
13. Srikant, R., Agrawal, R.: Mining Generalized Association rules. In: Proc. Of the 21st Int'l Conf. on Very Large Databases, Zurich, Switzerland, pp. 407–419 (1995)
14. Cakmak, A., Ozsoyoglu, G.: Taxonomy superimposed graph mining. In: Proceedings of the 11th Int'l Conf. on Extending Database Technology, Advances in Database Technology. ACM Int'l Conf. Proceeding Series, vol. 261, pp. 217–228 (2008)
15. Silvescu, A., Caragea, D., Atramentov, A.: Graph Databases Artificial Intelligence Research Laboratory. Department of Computer Science Iowa State University Ames, Iowa (2007)
16. Priyadarshini, S., Mishra, D.: A Hybridized Graph Mining Approach. In: Das, V.V., Vijaykumar, R. (eds.) ICT 2010. Communications in Computer and Information Science, vol. 101, pp. 356–361. Springer, Heidelberg (2010)
17. Pan, H., Tan, X., Han, Q., Yin, G.: An Effective Algorithm Based on Association Graph and Matrix for Mining Association Rules. In: Proceeding of IEEE 2010. IEEE PRESS (2010) 978-1-4244–6977-2/10
18. Muthukumar, A., Nadarajan, R.: Efficient and Scalable Partition based Algorithms for Mining Association Rules. Academic Open Internet Journal 19 (2006) ISSN 1311–4360
19. Balaji Raja, N., Balakrishnan, G.: Evaluation of Rule Likeness Measures for a Learning Environment. In: Proceedings of the ICOREM Int'l Conf., pp. 1682–1690. ICOREM (2009)
20. Balaji Raja, N., Balakrishnan, G.: A Model of Algorithmic Approach to Itemsets Using Association Rules. In: Proceedings of IEEE, ICECT 2011, pp. 6–10. IEEE PRESS (2011) 978-1-4244-8677-9