Optimization of Association Rule Learning in Distributed Database using Clustering Technique

Mr. Neeraj Raheja[1] Ravish Kumar[2]

[1]Computer Science Department, Assistant Professor, Maharishi Markandeshwar Engineering College/Maharishi Markandeshwar University, Mullana, Ambala, India
Neeraj_raheja2003@yahoo.co.in
[2]Computer Science Department, Research Scholar, Maharishi Markandeshwar Engineering College/Maharishi Markandeshwar University, Mullana, Ambala, India
Ravish.choudhary88@live.in

Abstract- Association rule mining is a way to find interesting associations among different large sets of data item. Apriori is the best known algorithm to mine the association rules. In this dissertation, clustering technique is used to improve the computational time of mining association rules in databases using Access data. Clusters are used to improve the performance of computer. Clusters are responsible for finding the frequent k item sets; hence lot of work is performed in parallel, thus decreasing the Computation time. This parallel nature of clusters is exploited to decrease the computation time in mining of data and also it reduces the bottleneck in the central site. Since after mining of data, there will be explosion of number of results and determining most frequent item sets will be difficult, so item sets are divided into two groups' namely-globally frequent item sets and locally frequent item sets.

Index Terms- Apriori Algorithm, Association Rule Learning, Clustering of Databases, Computation Time, Locally and Globally Frequent Items

I. INTRODUCTION

Data mining, also known as knowledge discovery in databases, has been recognized as a new area for database research. The area can be defined as efficiently discovering interesting rules from large collections of data. Association rule mining (ARM) has attracted tremendous interest among data mining researchers. ARM is an undirected or unsupervised data mining technique which works on variable length data, and produces clear and understandable results. The problem of mining association rules was introduced in [2]. Given a set of transactions, where each transaction is a set of items, an association rule is an expression X + Y, where X and Y are sets of items. The intuitive meaning of such a rule is that transactions in the database which contain the items in X tend to also contain the items in Y. An example of such a rule might be that 98% of customers who purchase tires and auto accessories also buy some automotive services; here 98% is called the confidence of the rule. The support of the rule X => Y is the percentage of transactions that contain both X and Y. The problem of mining association rules is to find all rules that satisfy a user-specified minimum support and minimum confidence. Applications include cross-marketing, attached mailing, catalog design, loss-leader analysis, store layout, and customer segmentation based on buying patterns.

Centralized data mining is used to discover useful patterns in organizations where each site of the organization, locally stores its ever increasing amount of day-to-day data. Data from these organizations are not only distributed over various locations but also vertically fragmented, making it difficult to combine them in a central location. Distributed data mining has thus emerged as an active subarea of data mining research. In this paper optimization of Association Rule Mining in distributed database has been done with the use of clustering technique.

II. PROBLEM STATEMENT

Frequent item sets from different databases come to a global database. Since there are so many databases through which frequent data goes to the global database so this increases the number of messages that need to be passed so as to find frequent k item set. Clustering technique is not used to group together a number of databases and thus causing a bottleneck around central site. The major problem with frequent set mining methods is the explosion of the number of results and so it is difficult to find the most interesting frequent item sets and so the concept of locally frequent item sets has been highlighted in this paper.

III. RELATED WORKS

Three parallel algorithms for mining association rules [2], an important data mining problem has been arisen in this paper. These algorithms have been designed to understand the performance implications of a spectrum of trade-offs between computation, communication, memory usage, synchronization, and use of problem-specific information in parallel data mining [11]. Fast Distributed Mining of association rules, which generates a small number of candidate sets and substantially reduces the number of messages to be passed at mining association rules [3].

The enormity and high dimensionality of datasets typically available as input to problem of association rule discovery, makes it an ideal problem for solving on multiple processors in parallel. The primary reasons are the memory and CPU speed limitations faced by single processors [1]. Algorithms for mining association rules from relational data have been well developed. Several query languages have been proposed, to assist association rule mining [18]. The topic of mining XML data has received little attention, as the data mining community has focused on the development of techniques for extracting common structure from heterogeneous XML data. For instance, [19] has proposed an algorithm to construct a frequent tree by finding common sub trees embedded in the heterogeneous XML data. The PADMA system [21] is a document analysis tool working on a distributed environment, based on cooperative agents. It works without any relational database underneath. Instead, there are PADMA agents that perform several operations with the information extracted from the documents.

A. ASSOCIATION ANALYSIS

Association Rule Analysis is useful for discovering interesting relationships hidden in large datasets. The uncovered relationships can be represented in the form of association rules or sets of frequent items.

Table 1. An example of market basket transactions.

TID	Items
1	{Bread, Milk}
2	{Bread, Diapers, Beer, Eggs}
3	{Milk, Diapers, Beer, Cola}
4	{Bread, Milk, Diapers, Beer}
5	{Bread, Milk, Diapers, Cola}

For example, the following rule can be extracted from the data set shown in Table 1:

{Diapers} -> {Beer}.

The rule suggests that a strong relationship exists between the sale of diapers and beer because many customers who buy diapers also buy beer. Retailers can use this type of rules to help them identify new opportunities for cross-selling their products to the customers.

Besides market basket data, association analysis is also applicable to other application domains such as bioinformatics, medical diagnosis, web mining, etc. For example, in the analysis of Earth science data, the association patterns may reveal interesting connections among the ocean, land, and atmospheric processes. There are two key issues that need to be addressed when applying association analysis to market basket data. First, discovering patterns from a large transaction data set can be computationally expensive. Second, some of the discovered patterns are potentially spurious because they may happen simply by chance.

B. ITEMSET AND SUPPORT COUNT

Let $I = \{i_1,i_2,.....,i_d\}$ be the set of all items in a market basket data and $T = \{t_1,t_2,....,t_N\}$ be the set of all transactions. Each transaction t_i contains a subset of items chosen from *I*. In association analysis, a collection of zero or more items is termed an item set. If an item set contains *k* items, it is called a *k*-item set. For instance, {Beer, Diapers, Milk} is an example of a 3-itemset. The null (or empty) set is an item set that does not contain any items.

The transaction width is defined as the number of items present in a transaction. A transaction t_j is said to contain an item set X if X is a subset of t_j. An important property of an item set is its support count, which refers to the number of transactions that contain a particular item set.

C. ASSOCIATION RULE

An association rule is an implication expression of the form $X \rightarrow Y$, where X and Y are disjoint item sets, i.e., $X \cap Y = \emptyset$. The strength of an association rule can be measured in terms of its support and confidence. Support determines how often a rule is applicable to a given data set, while confidence determines how frequently items in Y appear in transactions that contain X. The formal definitions of these metrics are:

Support, s $(X \rightarrow Y) = (X U Y)/N$
Confidence, c $(X \rightarrow Y) = (X U Y)/(X)$

Support is an important measure because a rule that has very low support may occur simply by chance. A low support rule is also likely to be uninteresting from a business perspective because it may not be profitable to promote items that customers seldom buy together. For these reasons, support is often used to eliminate uninteresting rules. Confidence, on the other hand, measures the reliability of the inference made by a rule. For a given rule $X \rightarrow Y$, the higher the confidence, the more likely it is for Y to be present in transactions that contain X. It also provides an estimate of the conditional probability of Y given X.

D. APRIORI ALGORITHM

Aprioiri is the first association rule mining algorithm that pioneered the use of support-based pruning to systematically control the exponential growth of candidate item sets. Apriori uses a "bottom up" approach, where frequent subsets are extended one item at a time (a step known as *candidate generation*), and groups of candidates are tested against the data. Apriori uses breadth-first search and a Hash tree structure to count candidate item sets efficiently. It generates candidate item sets of length K from item sets of length K-1. Then it prunes the candidates which have an infrequent sub pattern. According to the downward closure lemma, the candidate set contains all frequent K-length item sets. After that, it scans the transaction database to determine frequent item sets among the candidates.

The pseudo code for the algorithm is given below for a transaction database T and a support threshold of €. Usual set theoretic notation is employed; though note that T is a multiset. C_k is the candidate set for level k. Generate () algorithm is assumed to generate the candidate sets from the large item sets of the preceding level, heeding the downward closure lemma. Count[c] accesses a field of the data structure that represents candidate set c, which is initially assumed to be zero. The most important part of the implementation is the data structure used for storing the candidate sets, and counting their frequencies.

```
$L_1$ = {frequent items}
for (k=2; $L_{k-1}$=Ø; k++) do begin
$C_k$ = candidates generated from $L_{k-1}$ (that is: Cartesian product $L_{k-1}$ * $L_{k-1}$ and eliminating any k-1 size item set that is not frequent);
for each transaction t in database do
    increment the count of all candidates in $C_k$ that are contained in it
$L_k$ = candidates in $C_k$ with minimum support
end
```

```
Apriori (T,ϵ)
    $L_1 \leftarrow$ {large 1-item sets}
    $k \leftarrow 2$
    while $L_{k-1} \neq \emptyset$
      $C_k \leftarrow \{ c | c = a U \{b\} \wedge a \in L_{k-1} \wedge b \in U L_{k-1} \wedge b \neq a \}$
       for transactions $t \in T$
         $C_t \leftarrow \{ c | c \in C_k \wedge c \in t \}$
         for candidates $c \in C_t$
             count [c] $\leftarrow$ count[c] +1
       $L_k \leftarrow \{ c | c \in C_k \wedge count[c] \geq \epsilon \}$
       $k \leftarrow k+1$
         return $U_k L_k$
```

E. ARCHITECTURE

Proposed work is about optimizing the association rule mining in distributing technique using clustering technique. The work will be implemented in Java and the databases will be created in MS-Access. The data will be filled with pseudo entries that can be related to the data in the data warehouse of a super market. In this clustering of the Databases will be done so that the responsibility of finding a frequent k item set can be distributed over clusters which will increase the response time as well as decrease the number of messages need be passed and thus avoiding the bottleneck around central site. To solve the problem of optimizing association rule learning in data warehouses above architecture (Fig. 1) is proposed.

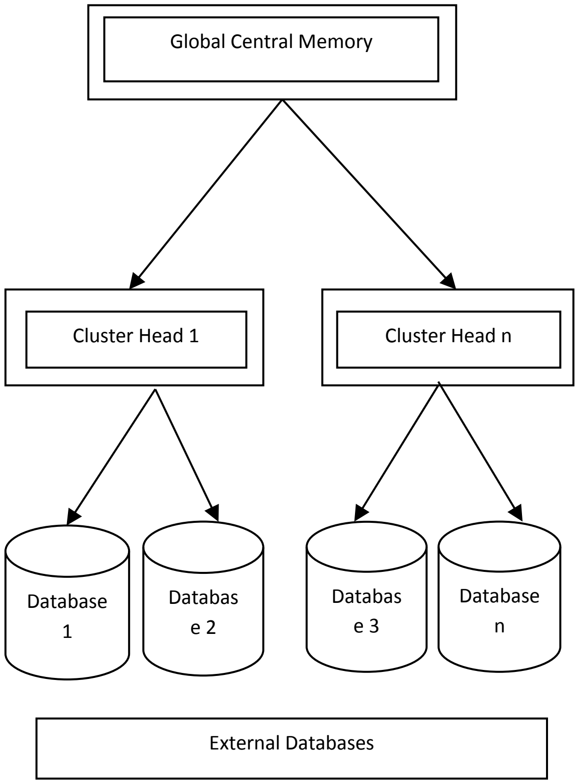

Figure 1: Architecture of Optimization of Association Rule Learning in Distributed Database

F. ALGORITHM

```
STEP 1: Input: Number of Clusters, Support Count
        Output: frequent k item sets
STEP 2: assign data warehouse to appropriate clusters
STEP 3: for each cluster
                a. find frequent k item sets
                b. store the item sets in the cluster Head
STEP 4: for each cluster Head
                For each item I in item sets
                If I ϵ more than n/2 clusters where n is number of clusters
                Set I as Globally Frequent Item sets
                                Else
Set I as Locally Frequent Item sets
                          End if
STEP 5: Display Globally and locally frequent item sets
STEP 8:         End of Algorithm
```

In this algorithm, the user decides the number of clusters and the support count. Then the nodes will be shown in their respective clusters. Then Apriori algorithm will be applied to each database and the frequent k item sets obtained from the database is stored in cluster head. An item set is stored in cluster head if more than n/2 (n is number of databases in cluster) possess that item set. This is known as majority rule. After each cluster has found its frequent k item sets, next step is to store frequent k item sets in global memory. Only those item sets are globally frequent which satisfies majority rule otherwise item sets will be store in locally frequent.

IV. RESULTS

Optimization of Association rule learning in Distributed Systems uses the approach of clustering various distributed nodes into a single cluster. Initially number of clusters is decided and accordingly databases are added to relevant clusters. Now clusters can perform in parallel to mine the association rules from the database. The mining is carried for each cluster and results are then stored in global memory thus avoiding a bottleneck around the global memory.

Proposed technique is implemented using java language and based on the results of the implementation, following results are concluded:

1. Computation Time: Proposed technique provides lesser Computation time than tradition method. This can be explained by the fact that since traditional techniques do not use the clustering approach, hence frequent set from all the databases are stored directly in the global memory, the global memory then finds frequent k item sets from this data. There is no concept of parallelism in this approach while in the proposed technique parallelism is achieved by the concept of clusters. Clusters are responsible for finding the frequent k item sets; hence lot of work is performed in parallel, thus decreasing the Computation time.
2. Introduction of Locally Frequent Associations: In the proposed technique, concept of locally frequent data is highlighted. This concept is very useful in certain scenarios like, super market. Based on locally frequent associations, decisions can be taken. This fact is only possible with the clustering technique.

The following are the two graphs which are obtained after implementing clustering technique in distributed databases.

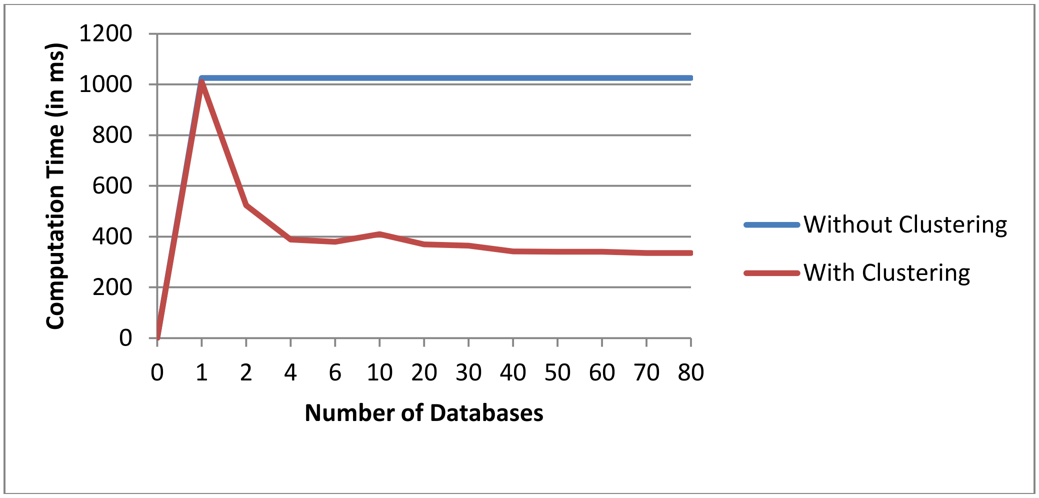

Figure 2: Computation Time versus Number of Databases

Figure 2 shows the computation time difference between “with clustering” and “without clustering” techniques. From this graph, it is clear that the computation time of “with clustering” technique is much low than “without clustering” technique.

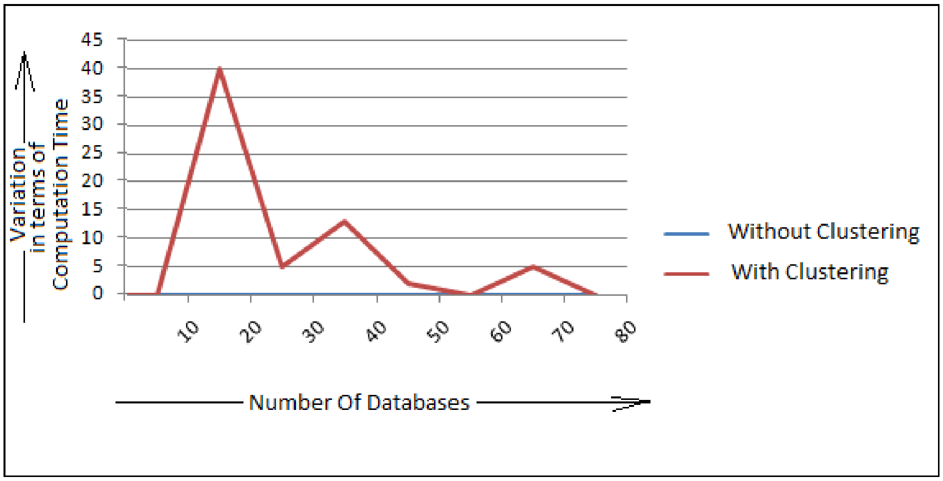

Figure 3: Variation Time in terms of Computation Time versus Number of Databases

Figure 3 shows the variation of time in both the techniques. It is clear from the graph that there will be variation in “with clustering” technique as the computation time in this techniques also varies from database to database.

V. CONCLUSION

Optimization of Association rule learning in Distributed Databases using clustering technique has proved to be a useful technique for mining, since it has shown a considerable amount of decrease in the computational time. This can prove to be very useful since mining usually involves very large data warehouses and thus clustering the warehouses into specific clusters can show a great improvement in computation time. Without clustering, computation time will always be high and so to reduce the computation time clustering technique has been applied. The computational time of traditional approach with proposed approach will vary as the number of databases increases and thus showing major difference between traditional approach and proposed approach. The major problem with frequent set mining methods is the explosion of the number of results and so it is difficult to find the most interesting frequent item sets and so the concept of locally frequent item sets has been highlighted in this dissertation. The future enhancement of this is to add global caching as caching can be used since data in warehouse tend to change a little over time. Techniques of clustering the databases can be debated upon, since more efficient the division of sites, more efficient will be association rules.

ACKNOWLEDGMENT

At the start of every major task, initially difficulties seem insurmountable. But gradually, as we begin treading on the chosen path with honesty and perseverance, we find that God is there to help us at each step. Towards the completion of my thesis, it’s a pleasurable aspect that I have now the break to communicate my thankfulness to all those who have been influential in some way or the other in the flourishing completion of this project. To begin with I would like to acknowledge **The Almighty God** who has been the prime mover in all my efforts. I owe my deepest sincere thankfulness to him for giving me the strength, the spirit, the will, the courage to work in the right direction to achieve all the goals of my life.

First of all, I would like to thank my supervisor Mr. Neeraj Raheja for his enormous help in planning and executing the work in time. His untiring and seemingly unlimited support, his relational persona, focused guidance, constructive criticism and above all, his availability to guide me both when I was delighted with the research and when I was despaired of it, instigated me in putting my hundred percent efforts. He has always tried to instill in me the value of quality work which can only be achieved after putting in the best of our sincere efforts and hard work. His overly enthusiasm and integral view on research and her mission for providing “only high quality work and not less” has made a deep impression on me.

I would like to express a deep sense of gratitude and thanks profusely again to Dr. V.K. Katiyar, Prof. and Head, Department of Computer Engineering, MMEC, Mullana has been of great help in carrying out the present work and is acknowledged with reverential thanks.

Finally, I want to thank my parents, who have always taught me the value of sincere efforts and hard work and also I would like to thank all those whose direct and indirect support helped me in completing my thesis in time.

REFERENCES

1. Dr (Mrs).Sujni Paul, “AN OPTIMIZED DISTRIBUTED ASSOCIATION RULE MINING ALGORITHM IN PARALLEL AND DISTRIBUTED DATA MINING WITH XML DATA FOR IMPROVED RESPONSE TIME”, International Journal of Computer Science and Information Technology, Volume 2, Number 2, April 2010
2. R. Agrawal and R. Srikant , "Fast Algorithms for Mining Association Rules in Large Database,” Proc. 20th Int'l Conf. Very Large Databases (VLDB 94), Morgan Kaufmann, 1994,pp. 407-419.
3. D.W. Cheung , et al., "A Fast Distributed Algorithm for Mining Association Rules," Proc. Parallel and Distributed Information Systems, IEEE CS Press, 1996,pp. 31-42
4. A. Savasere , E. Omiecinski, and S.B. Navathe , "An Efficient Algorithm for Mining Association Rules in Large Databases,"Proc. 21st Int'l Conf. Very Large Databases (VLDB 94), Morgan Kaufmann, 1995, pp. 432-444
5. M.J. Zaki , "Scalable Algorithms for Association Mining,"IEEE Trans. Knowledge and Data Eng.,vol.12 no. 2, 2000,pp. 372-390;
6. J.S. Park , M. Chen, and P.S. Yu , "An Effective Hash Based Algorithm for Mining Association Rules,"Proc. 1995 ACM SIGMOD Int'l Conf. Management of Data , ACM Press, 1995, pp. 175-186.
7. M.J. Zaki , et al., Parallel Data Mining for Association Rules on Shared-Memory Multiprocessors , tech. report TR 618, Computer Science Dept., Univ. of Rochester, 1996.
8. D.W. Cheung , et al., "Efficient Mining of Association Rules in Distributed Databases,"IEEE Trans. Knowledge and Data Eng., vol. 8, no. 6, 1996,pp.911-922;

9. A. Schuster and R. Wolff , "Communication-Efficient Distributed Mining of Association Rules," Proc. ACM SIGMOD Int'l Conf. Management of Data, ACM Press, 2001,pp. 473-484.
10. M.J. Zaki , "Parallel and Distributed Association Mining: A Survey,"IEEE Concurrency, Oct.- Dec. 1999,pp. 14-25.
11. C.L. Blake and C.J. Merz , UCI Repository of Machine Learning Databases, Dept. of Information and Computer Science, University of California, Irvine, 1998;
12. T. Shintani and M. Kitsuregawa , "Hash-Based Parallel Algorithms for Mining Association Rules,"Proc. Conf. Parallel and Distributed Information Systems, IEEE CS Press, 1996, pp. 19-30;
13. C.C. Aggarwal and P.S. Yu , "A New Approach to Online Generation of Association Rules,"IEEE Trans. Knowledge and Data Eng. , vol. 13, no. 4, 2001,pp.527-540
14. Hillol Kargupta, Ilker Hamzaoglu, and Brian Stafford. “Scalable, distributed data mining-an agent architecture”, IEEE, vol 14, pp 231-237.
15. A. Termier, M.-C. Rousset, and M. Sebag. Mining XML data with frequent trees. In DBFusion Workshop’02,pp 87–96.
16. R. Meo, G. Psaila, and S. Ceri. A new SQLlike operator for mining association rules. In The VLDB Journal, pp 122–133, 1996.
17. Mafruz Zaman Ashrafi, David Taniar,Kate Smith, Monash University “ODAM: An Optimized Distributed Association Rule Mining Algorithm”, IEEE distributed systems online 1541-4922,vol. 5, no. 3; march 2004
18. T. Imielinski and A. Virmani. MSQL: A query language for database mining. 1999.
19. A. Termier, M.-C. Rousset, and M. Sebag. Mining XML data with frequent trees. In DBFusion Workshop’02, pages 87–96.
20. A. Prodromidis, P. Chan, and S. Stolfo. Chapter Meta-learning in distributed data mining systems: Issues and approaches. AAAI/MIT Press, 2000.