Association Rule Mining on Distributed Data

Pallavi Dubey

Abstract - Applications requiring large data processing, have two major problems, one a huge storage and its management and second processing time, as the amount of data increases. Distributed databases solve the first problem to a great extent but second problem increases. Since, current era is of networking and communication and people are interested in keeping large data on networks, therefore, researchers are proposing various algorithms to increase the throughput of output data over distributed databases. In my research, I am proposing a new algorithm to process large amount of data at the various servers and collecting the processed data on client machine as much as he/she is requiring. The data is kept in XML format, which allows processing it further, if needed.

The local copy of searched data is provided to the users if he/she requires it again, this allows making a proxy server where frequently searched items can be kept with the frequency of their access. This not only allows providing fast access to the data but will also provide to maintain list of frequently accessed data.

For accessing the data from the various servers, there are several methods such as mobile agents, direct networked access, client-server techniques Etc. I have used multithreaded environment to map various distributed servers to collect data. For processing of data at the server end, Apriori Algorithm has been applied to get the outputs, which are then sent to the client. At client data from various servers is collected and list of uncommon data is created which is then converted into XML data format. If the search is successful then user is allowed to store the search locally or at proxy server, this will reduce the future processing time of the same data search. In this paper an Optimized Distributed Association Rule mining algorithm for geographically distributed data is used in parallel and distributed environment so that it reduces communication costs. The response time is calculated in this environment using XML data.

Keywords - Association rules, Apriori algorithm, parallel and distributed data mining, Multiprocessing Environment, XML data, response time.

---

1. INTRODUCTION

Association rule mining (ARM) has become one of the core data mining tasks and 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. There are two dominant approaches for utilizing multiple Processors that have emerged; distributed memory in which each processor has a private memory; and shared memory in which all processors access common memory [5]. Shared memory architecture has many desirable properties. Each processor has direct and equal access to all memory in the system. Parallel programs are easy to implement On such a system. In distributed memory architecture each processor has its own local memory that can only be accessed directly by that processor [10]. For a processor to have access to data in the local memory of another processor a copy of the desired data element must be sent from one processor to the other through message passing. XML data are used with the Optimized Distributed Association Rule Mining Algorithm.

A Parallel application could be divided into number of tasks and executed concurrently on different processors in the system [9]. However the performance of a parallel application on a distributed system is mainly dependent on the allocation of the tasks comprising the application onto the available processors in the system.In different kinds of information databases, such as scientific data, medical data, financial data, and marketing transaction data; analysis and finding critical hidden information has been a focused area for researchers of data mining. How to effectively analyze and apply these data and find the critical hidden information from these databases, data mining technique has been the most widely discussed and frequently applied tool from recent decades. Although the data mining has been successfully applied in the areas of scientific analysis, business application, and medical research and its computational efficiency and accuracy are also improving, still manual works are required to complete the process of extraction. Association rule mining model among data mining several models, including Association rules, clustering and classification models, is the most widely applied method. The Apriori algorithm is the most representative algorithm for association rule mining. It consists of many modified algorithms that focus on improving its efficiency and accuracy. For the purpose of simulation, I have employed the database of Industries to assess the proposed algorithm.The rest of this study is organized as follows. Section 2 briefly presents the general background, while the proposed method is explained in Section 3. Sections 4 and 5 illustrate the computational results of the Industry database. The concluding remarks are finally made in Section 6.

2. LITERATURE REVIEW

Association Rule Mining: In data mining, association rule Learning is a popular and well researched method for discovering interesting relations between variables in large databases. It analyzes and present strong rules discovered in databases using different measures of interestingness. Based

on the concept of Strong, rules, Agrawal et al., introduced association rules for discovering regularities between products in

large scale transaction data recorded by point-of-sale (POS) systems in supermarkets.

For example, the rule found in the sales data of a supermarket would indicate that if a customer buys onions and potatoes together, he or she is likely to also buy burger. Such information can be used as the basis for decisions about marketing activities such as, e.g., promotional pricing or product placements. In addition to the above example from market basket analysis association rules are employed today in many application areas including Web usage mining, intrusion detection and bioinformatics. Three parallel algorithms for mining association rules [3], an important data mining problem is formulated in this paper. These algorithms have been designed to investigate and understand the performance implications of a spectrum of trade-offs between computation, communication, memory usage, synchronization, and the 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 [4].

Algorithms for mining association rules from relational data have been well developed. Several query languages have been proposed, to assist association rule mining such as [12], [13]. 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, [14] has proposed an algorithm to construct a frequent tree by finding common sub trees embedded in the heterogeneous XML data. On the other hand, some researchers focus on developing a standard model to represent the knowledge extracted from the data using XML. JAM [15] has been developed to gather information from sparse data sources and induce a global classification model. The PADMA system [16] 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 relational operations with the information extracted from the documents.

ASSOCIATION RULE MINING ALGORITHMS

An association rule is a rule which implies certain association relationships among a set of objects (such as ``occur together'' or ``one implies the other'') in a database. Given a set of transactions, where each transaction is a set of literals (called items), an **association rule** is an expression of the form X Y, where X and Y are sets of items. The intuitive meaning of such a rule is that transactions of the database which contain X tend to contain Y Association rule mining(ARM) is one of the data mining technique used to extract hidden knowledge from datasets that can be used by an organizations decision makers to improve overall profit.[2].

2.1 Apriori Algorithm

An association rule mining algorithm, Apriori has been developed for rule mining in large transaction databases by IBM's Quest project team [4]. An {item set} is a non-empty set of items.

They have decomposed the problem of mining association rules into two parts:

1. Find all combinations of items that have transaction support above minimum support. Call those combinations frequent item sets. Item.
2. Use the frequent item sets to generate the desired rules. The general idea is that if, say, ABCD and AB are
3. frequent item sets, and then we can determine if the Rule AB CD holds by computing the ratio

   r = support (ABCD)/support (AB).

The rule holds only if r >= minimum confidence. Note that the International Journal of Computer Science and Information Technology, Volume 2, Number 2, April 2010 90 rule will have minimum support because ABCD is frequent. The algorithm is highly scalable [8].

The Apriori algorithm used in Quest for finding all frequent item sets

is given below.

Distributed/parallel algorithms

Databases or data warehouses may store a huge amount of data to be mined. Mining association rules in such databases may require substantial processing power [7]. A possible solution to this problem can be a distributed system. [6]. Moreover, many large databases are distributed in nature which may make it more feasible to use distributed algorithms. Major cost of mining association rules is the computation of the set of large item sets in the database. Distributed computing of large item sets encounters some new problems. One may compute locally large Item sets easily, but a locally large item set may not be globally large. Since it is very expensive to broadcast the whole data set to other sites, one option is to broadcast all the counts of all the item sets, no matter locally large or small, to other sites. However, a database may contain enormous combinations of

item sets, and it will involve passing a huge number of messages.

A distributed data mining algorithm FDM (Fast Distributed Mining of association rules) has been proposed by [6].

Distributed Algorithms [17]

1. Distributed association rule learning
2. Collective decision tree learning
3. Collective PCA and PCA-based clustering
4. Distributed hierarchical clustering
5. Other distributed clustering algorithms
6. Collective Bayesian network learning

The Four Parallel Algorithms are

1. Count Distribution -- parallelizing the task of measuring the frequency of a pattern inside a database

2. Candidate Distribution -- parallelizing the task of generating longer patterns
3. Hybrid Count and Candidate Distribution -- a hybrid algorithm that tries to combine the strengths of the above algorithms

4. Sampling with Hybrid Count and Candidate Distribution -- an algorithm that tries to only use a sample of the database.

In a parallel data mining the main issues taken into account are

1. Load balancing
2. Minimizing communication
3. Overlapping communication and computation

2.2 OPTIMIZED DISTRIBUTED ASSOCIATION RULE MINING ALGORITHM

The performance of Apriori ARM algorithms degrades for various reasons. It requires {n} number of database scans to generate a frequent {n}-item set. Furthermore, it doesn't recognize transactions in the data set with identical item sets if that data set is not loaded into the main memory. Therefore, it unnecessarily occupies resources for repeatedly generating item sets from such identical transactions. For example, if a data set has 10 identical transactions, the Apriori algorithm not only enumerates the same candidate item sets 10 times but also updates the support counts for those candidate item sets 10 times for each iteration. Moreover, directly loading a raw data set into the main memory won't find a significant number of identical transactions because each transaction of a raw data set contains both frequent and infrequent items. To overcome these problems, we don't generate candidate support counts from the raw data set after the first pass. This technique reduces the average transaction length

International Journal of Computer Science and Information Technology, Volume 2, Number 2, April 2010 93 data set size significantly, so we can accumulate more transactions in the main memory. The number of items in the data set might be large, but only a few will satisfy the support threshold.

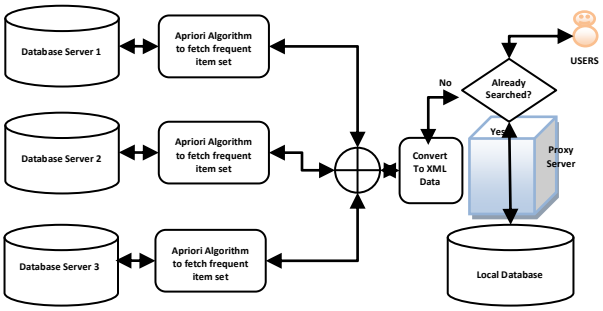
Figure 1. Basic Format of XML Data

```
<transactions>
        <transaction id=1>
                <items>
                        <item> i1</item>
                        <item> i4</item>
                        <item> i7</item>
                </items>
        </transaction>
        <transaction id=2>
                <items>
                        <item> i2</item>
                        <item> i3</item>
                        <item> i5</item>
                </items>
        </transaction>
        <transaction id=3>
                <items>

<item> i1</item>
                        <item> i3</item>
                        <item> i7</item>
                </items>
        </transaction>


        <transaction id=4>
                <items>
                        <item> i2</item>
                        <item> i5</item>
                </items>
        </transaction>
        <transaction id=5>
                <items>
                        <item> i1</item>
                        <item> i5</item>
                </items>
        </transaction>
        <transactions>
```

3. Proposed Algorithm

The algorithm developed to provide the distributed data at a very fast rate to the users involve flow of processing of data as follows:

Figure 2. Flow Chart to show the proposed algorithm



The algorithm works in following steps:

1. User demands the data from the interface provided.
2. Data demanded is transferred to the proxy server, where it is first checked in the local database for availability, if the data is available, then it is provided to the user and frequency of data is increased by one.
3. If data is not available in local database then it is transferred to the various
4. Distributed databases using multithreaded environment for parallel processing.
5. The various servers send the desired results to the proxy server, where it is merged together to find the uncommon item set for the searched value Item User / Proxy Server Agent is allowed to store the results locally so that Future searching of the same value will not take longer time.
6. Proxy server Agent / User has been provided with the facility of setting Support threshold percent before processing and also provide the facility of searching for more than item at a time and in a fast speed of searching for single value and more than one value less amount of time is needed.

4. RESULTS & DISCUSSION

For implementing the proposed algorithm and generating various results JAVA language is used to create a simulation environment. An INDUSTRY database has been chosen to find the stock price values for various items in the databases. Three different copies of the database is created for simulation purposes

Figure 3. Simulation Environment Snap shot 1

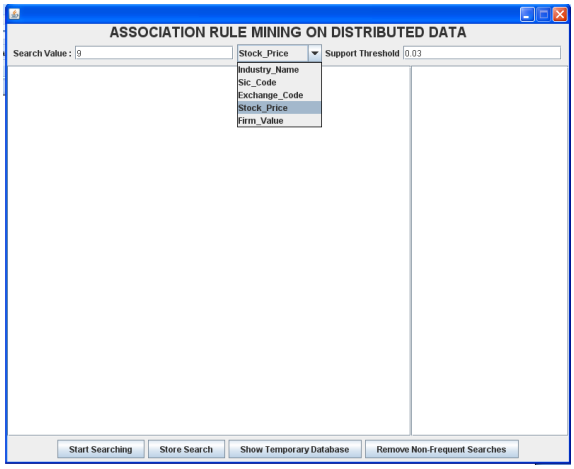


Figure 4. Simulation Environment Snap shot

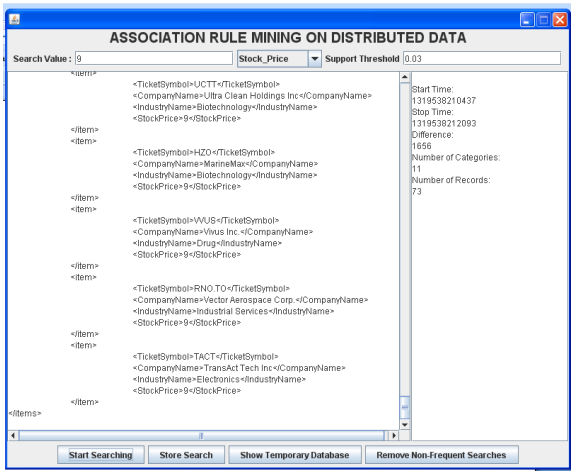
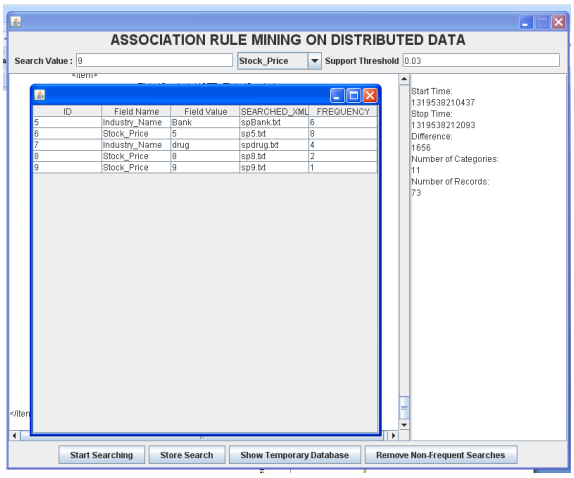


Figure 5. Simulation Environment Snap shot



to show the processing of the distributed databases.JAVA provides excellent thread model processing for mapping of the distributed databases and parallel systems. Simulation environment created is as shown in the figures 3-5.After running the simulation for the various stock price values and support threshold values following data have been gathered:

Table 1. Data collected Threshold Vs Time

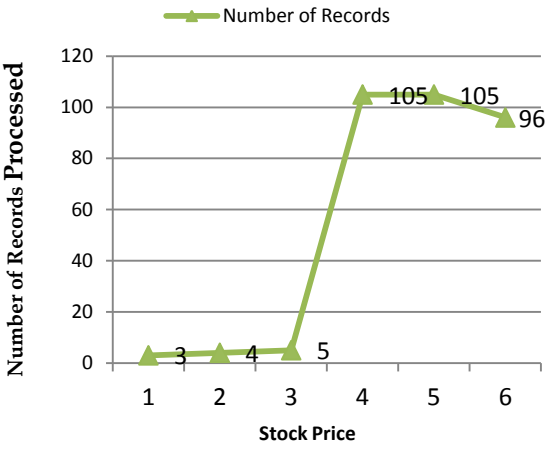
Threshold	Time Taken
0.020	140
0.025	172
0.030	235
0.015	125
0.010	78
0.005	62

Table 2. Data collected no of Records processed Vs Time

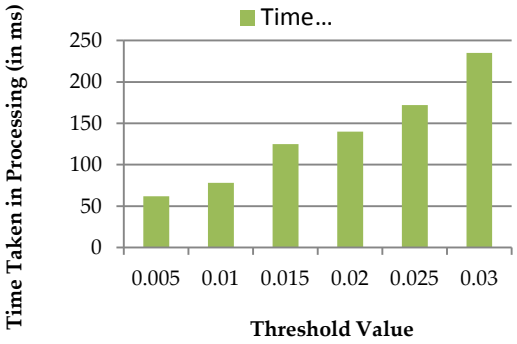
Stock Price	Number of Records	Time Taken
1	3	500
2	4	469
3	5	485
4	105	547
5	105	594
6	96	532

. Graphs plotted from the above collected data shows the various results as follows:

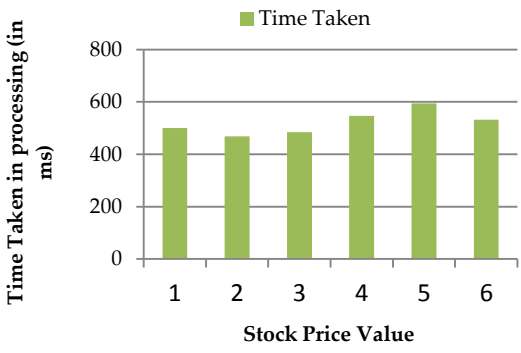
No of records Vs stock price

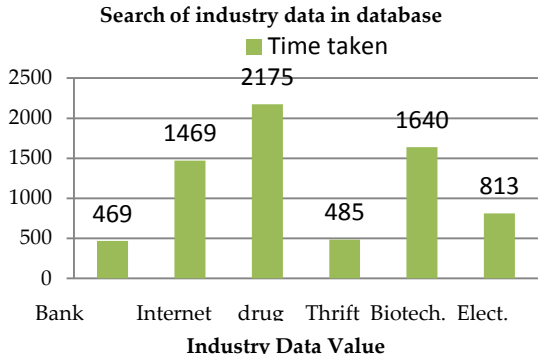


Effect of support threshold



Search of Stock Price in Industry Database





5.CONCLUSION & FUTURE ENHANCEMENTS

Association rule mining is an important performance association rule mining on data. The Optimized Distributed Association Mining Algorithm is used for the mining process distributed environment. The response time with the communication and computation factors are considered to achieve an improved response time, number of processors in a distributed environment.

As the mining process is done in parallel an optimal solution is obtained. The various graphs show the processing time as expected and generate the results as per the requirements of the users. Fast response time as shown in the graphs shows that the proposed algorithm generates the results as required. The Future enhancement of this is to work around on proxy server to allow users to access new data searched even when the data is found locally, it is required because searched data may vary at

REFERENCES

[1] Dr (Mrs).Sujni Paul, Associate Professor, Department of Computer Applications, Karunya University, Coimbatore 641114 , Tamil Nadu, India

[2] R. Agrawal and R. Srikant , "Fast Algorithms for Mining Association Rules in Large Database," *Conf. Very Large Databases* (VLDB 94), Morgan

[3] R. Agrawal and J.C. Shafer , "Parallel Mining of Association Rules," Distributed Systems Online March 2004

[4] D.W. Cheung , et al., "A Fast Dis *Distributed Information Systems,* IEEE CS Press, 1996,pp. 31

[5] 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*

[6] J. Han , J. Pei, and Y. Yin , "Mining Frequent Patterns without Candidate Generation," *Int'l. Conf. Management of Data* , ACM Press, 2000,pp. 1

[7] Kaufmann, 1994,pp. 407-419. *IEEE Tran. Knowledge and Data Eng.* , vol. 8, no. 6, 1996,pp. 962-969;. Distributed Algorithm for Mining Association Rules," 31-42; (VLDB 94),

[8] Morgan Kaufmann, 1995, pp. 432 *Proc. ACM SIGMOD* 1-12. 2010 99 is the time to In our approach, result. in a parallel and performance *Proc. 20th Int'l* 16 IEEE tributed *Proc. Parallel and* 432-444. International Journal of Computer Science and Information Technology, Volume 2, Number 2, April 2010 100

[9] 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.

[10] 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;

[11] A. Schuster and R.op Wolff , "Communication-Efficient Distributed Mining of Association Rules," *Proc. ACM SIGMOD Int'l Conf. Management of Data,* ACM Press, 2001,pp. 473-484.

[12] T. Imielinski and A. Virmani. MSQL: A query language for database mining. 1999.

[13] R. Meo, G. Psaila, and S. Ceri. A new SQLlike operator for mining association rules. In The VLDB Journal, pages 122–133, 1996.

[14] A. Termier, M.-C. Rousset, and M. Sebag. Mining XML data with frequent trees. In DBFusion Workshop'02, pages 87–96.

[15] A. Prodromidis, P. Chan, and S. Stolfo. Chapter Meta learning in distributed data mining systems: Issues and approaches. AAAI/MIT Press, 2000.

[16] Hillol Kargupta, Ilker Hamzaoglu, and Brian Stafford. Scalable, distributed data mining-an agent architecture. In Heckerman et al. [8], page 211.

[17] Albert Y. Zomaya, Tarek El-Ghazawi, Ophir Frieder, "Parallel and Distributed Computing for Data Mining", IEEE Concurrency, 1999. International Journal of Computer Science and Information Technology, Volume 2, Number 2, April