Improving Association Rule based Data Mining Algorithms with Agents Technology in Distributed Environment

P. N Santhosh Kumar, Dr C. Sunil Kumar, Dr C. Venugopal

Assistant Professor in ECM, SNIST (An Autonomous Institution), Hyderabad, A.P., India
Professor in ECM, SNIST (An Autonomous Institution), Hyderabad, A.P., India
Professor in ECM, SNIST (An Autonomous Institution), Hyderabad, A.P., India

Abstract- Applying distribution in the form of agents technology and improving association rule based data mining algorithms, agents are the best for doing the continuous data mining efficiently reducing network load and carrying the code to remote locations and the types of mobile agent which having the energy of moving from one place to another itself ,and interacting with the other mining agents for the purpose of data mining as component based communication, so working with communication agents for improvement of apriori algorithm for quality. By proposing architecture for improvement of apriori mining in distributed environment, this architecture can facilitate the services of instantaneous real time messaging; this study includes the issue of constantly executing queries on continuous data in heterogeneous platforms or environments.

I. Introduction

The Apriori Algorithm is used to find Frequent Item Sets (FISs) Using Candidate Generation Apriori is an influential algorithm for mining FIS for Boolean association rules. The name of the algorithm is based on the fact that the algorithm uses prior knowledge of FIS properties. Apriori utilizes an iterative approach known as a level-wise search, where k-item sets are used to explore (k+1)-item sets. First, the set of frequent 1-item sets is found. This set is denoted L1. L1 is used to find L2, the set of frequent 2-item sets, which is used to find L3, and so on, until no more frequent k-item sets can be found. The finding of each Lk requires one full scan of the local relational databases. With more importance of bigger data files distributed over widereanetwork (WAN), where limitations of efficient bandwidth and software tools, forced the development of distributed data mining (DDM). DDM is expected to partial analyze data partially at individual sites and then to send the outcome as partial result to other sites where it is, Partially at individual sites and then to send the outcome as partial result to other sites where it is sometimes required to be unified for achieving global result. In order to support distributed architectures, there are two architectures mainly client server and agents communicative agents. In this project we explore the capabilities of mobile agents in a DDM.

Consider these typical situations where, central data server that has to collect data from several computers; like WWW search engine collects data from web servers all over the world. Central information server in your company collects data from different departments and data mining on a large distributed database [**1, 9**]. Conventional Data Collection (CDC), Decentralized Data Collection (DDC), Data collection with communication agents (DCCM) are three preferred solutions. CDC collects data from several computers with the following Disadvantages like the central server needs all the data from the other computers before it can do some processing and, in DDC all computers run a kind of distributed search engine, for example Harvest. The local search engines process data locally and transfer the results to central server. And Disadvantages are, lot of maintenance for the local search engines is needed, when a new version of the search engine comes up it must be installed on every local server. Where as in DCCM, with the help of communication agents (CA), it travels around the distributed environment for mining. At each terminal it processes the data and sends the results back to the central server provides Low network traffic because the

agents do data local processing [6]. For identifying better algorithm and suitable mining architecture in distributed environments. The objective of the paper is to develop architecture to support Distributed Data Mining (DDM), which can be used to extract hidden predictive information from large databases, henceforth will be a great potential use to help companies. The focus is on the most important information on their data repositories.

The existing data mining algorithms for distributed data are of communication intensive [2]. Many algorithms for data mining have been proposed for a data at a local host based data repository, and applied some of them are implemented at multiple locations with little bit improvement, in terms of efficiency of these algorithms as a part of quality but complexity of algorithms are not efficient in distributed environment are not addressed, as data on the web/network are distributed by very of its nature. As a consequence, both new architectures and new algorithms are needed to merge together.

Mobile agents carries state attributes and code which defines agent's behavior like (when and where to move , what to process there , special type of agents is 'communication agent 'couriers messages back and forth between clients residing on various network nodes as multi server multi clients based computation environment. So distributed data mining is performed efficiently with the association of agents, where communication in between these agents achieved in the following methodology. The control of communication agents are managed by using java classes of Aglet tool kit, some important calls are briefed, those are Future Reply Class Evaluates Weather a Reply will Be Given to a Message, an Aglet Can Perform another Task while by calling proxy.sendAsyncMessage; Any aglet that wants to communicate with other aglets has to first obtain the proxy object. And use the following calls. getAgletInfo, communication achieved by exchanging aglets of Message class. Agletproxy Class is Responsible for Sending and receiving by sendMessage of Aglet proxy class.

II. Literature Survey

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 [6, 7 and 8]. It analyzes and present strong rules discovered in databases using different measures of interestingness. Based on the concept of strong association rules for discovering regularities between products in large scale transaction data recorded by point – of - sale (POS) systems in supermarkets are introduced.

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, an important data mining problem is formulated in this paper [3]. 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. 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. Algorithms for mining association rules from relational data have been well developed. Several query languages have been proposed, to assist association rule mining. 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 has been developed to gather information from sparse data sources and induce a global classification model. The PADMA system 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.

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.

III. Apriori Analyses in Distributed Data Mining

However the data parallelism algorithms need more memory at each remote site for storing all candidates for each scan, performance will be degraded if not providing much memory. The task parallelism algorithms can avoid this type of degrading. The task distribution may work where data distribution may not work [4]. Henceforth estimated approach that is using mobile agents for task distribution will give efficient results. Investigating suitability of Apriori algorithm for parallel approach, proposed 4 parallel algorithms based on Apriori; speed up mining of frequent item sets.

Fourth type The Candidate Distribution Mining (CDM) algorithm parallelizes the task of generating longer patterns and load balancing algorithm that reduces synchronization between the processors and segments. The database is based upon different transaction patterns. These parallel algorithms were tested among each other and CD had the best performance against the Apriori algorithm. Its overhead is less than 7.5% when compared with Apriori by tried to make DDM and CDM scalable by Hybrid Distribution (HD) algorithm respectively.CDM addresses the issues of communication solves overhead and redundant computation in by using aggregate memory to partition candidates and move data efficiently.CDM improves over by dynamically partitions the candidate set to maintain good load balance. Experiment output results show that the response time of CDM is 4.4 times less than DDM on a 32-processors system and HD is 9.5% better than CDM on 128 processors. the following graph shows comparison of the running time of above four algorithms.; so it is considered as stable in DDM.and the proposed algorithm is given in Figure 3.1.

A. Apriori Algorithm

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

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

- Find all combinations of items that have transaction support above minimum support. Call those combinations frequent itemsets.
- Use the frequent itemsets to generate the desired rules. The general idea is that if, say, ABCD and AB are frequent itemsets, 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 assurance. Note that the rule will have minimum support because ABCD is frequent. The algorithm is highly scalable. The Apriori algorithm used in Quest for finding all frequent itemsets is given below.

ALGORITHM Apriori_gen ():-

For all agents if (agents=true) {
I1 =item set A; I2=item set B;
Procedure combine () {
For all up to k-1 {

Select p.item1... I1.itemk-1, I2.itemk-1 from Lk-1I1, Lk-1I2
where I1.item1 = I2.item1 and I1.itemk-2 = I2.itemk-2 and I1.itemk-1 < I2.itemk-1 ;}}
If (agent=false)

// eliminate item sets such that some (k-1)-subset not in L_{k-1} for all c in Ck
Procedure eliminate () {
For all (k-1)-subsets s of c if (s is not in L_{k-1}) {eliminate c from c_k; break; } }

Figure 1: Algorithm for Apriori_Agent

It makes numerous passes over the database. In the first pass, the algorithm basically counts item occurrences to determine the frequent 1-itemsets (itemsets with 1 item). A successive pass say pass k, consists of two phases. First, the frequent itemsets I_{k-1} (the set of all frequent (k-1)-itemsets) found in the $(k-1)^{Th}$ pass are used to generate the candidate itemsets C_k, using the apriori-gen () function. This function first joins L_{k-1} with I_{k-1}, the joining state being that the lexicographically ordered first k-1 items are the same. Next, it deletes all those itemsets from the join result that have some (k-1)-subset that is not in I_{k-1} yielding C_k.

The algorithm now scans the database. For each transaction, it determines which of the candidates in C_k are contained in the transaction using a hash-tree data structure and increments the count of those candidates. At the end of the pass, C_k is examined to determine which of the candidates frequent, yielding I_k. The algorithm terminates when I_k becomes empty.

IV. Architecture for Web Services in Distributed Data Mining

The Proposed architecture uses XML standards and web services which are important **[5]**. They keep data mining algorithms as web services, which are invoked and utilized by different knowledge discovery applications located in distributed, locations. The main components and the functionalities of these components are described in the sequential order.

Describing components from top-left the webbot is a very fastest and reliable web walker with support for regular expressions, sql logging files continuousdata, web log files collected in parallel,webserver Log files are downloaded and sessionizer generates a LOGML file, Integration Engine(IE) which is the part of Data Warehouse (DW) functionality which is suited for preprocessing of data at remote sites, after integration finally loading into database and later generating patterns in the form of graphs, User sessions from web logs are extracted for studying and analyzing sequences of similarity, so distribute mined by using DDM, Here using the logic in the following manner, frequent contiguous sequences with a given minimum support. These are imported into a database, the minimal frequent sequences are suppressed, and Different queries are faced Algorithms. Apriori and AprioriTid as combination of Apriori _agent (); against this data according to some criteria of minsupport of each pattern. Different fragmented results obtained by different communication agents are to be merged for obtaining target output. Predictive Model Markup Language (PMML) is an XML-based language which follows a very intuitive structure to describe data pre- and post-processing as well passing models as input to the another algorithms .PMML is used to transform raw data into meaningful features It wouldn't be complete to describe web services without mentioning the SOAP protocol **[10, 11 and 12]**. SOAP is not really simple protocol and "object" has nothing to do with the protocol, there is no importance to understand SOAP as it is transparent to you unless you deal with related low-level programming. The web service architecture is shown in the Figure 2.

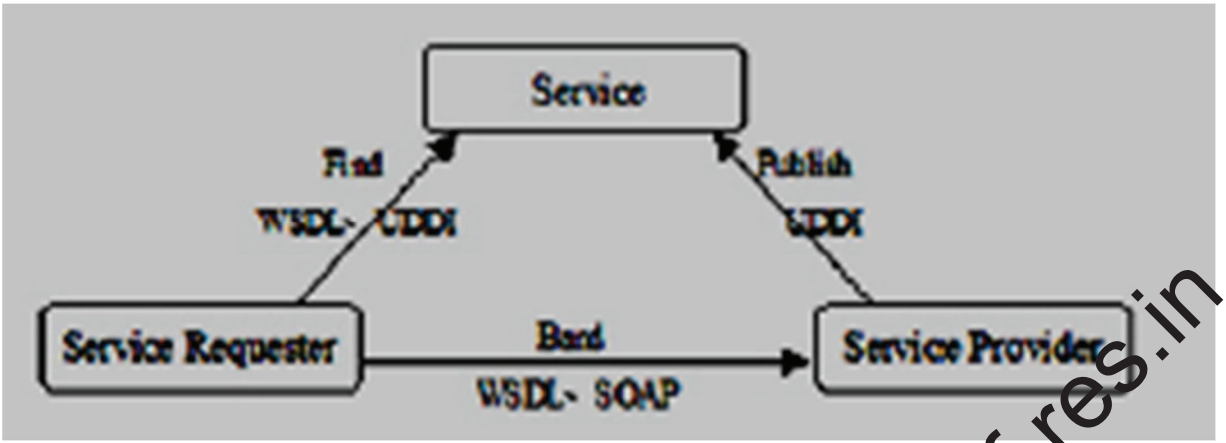

Figure 2: Web Service Architecture and related standards

V. Implementation with Results

The implementation of Apriori_Agent () algorithm with Aglet class, AgletMessageclass and Aglet Proxy classes and writing..\cnf\aglets.props and build XML files, build successful after running ANT command; Tahiti server automatically started after running C:\aglets\bin\agletsd -f.\ aglets. props command, now dispatching the mining aglet to another host, the following graphical user interface created and our data mining agent ready to do apriori mining on the Web Server Log files which are downloaded and processed through a sessionizer and save that file as LOGML ,another host located in distributed environment may asked to supply support values and the task of apriori mining is completed at remote host as shown in Figure 3 in the similar way any remote host supply datasets like log files, weblogs,LOGMLs,data.txt as input to our Distributed Data Mining (DDM) with the help of aglets which are event based mechanisms.

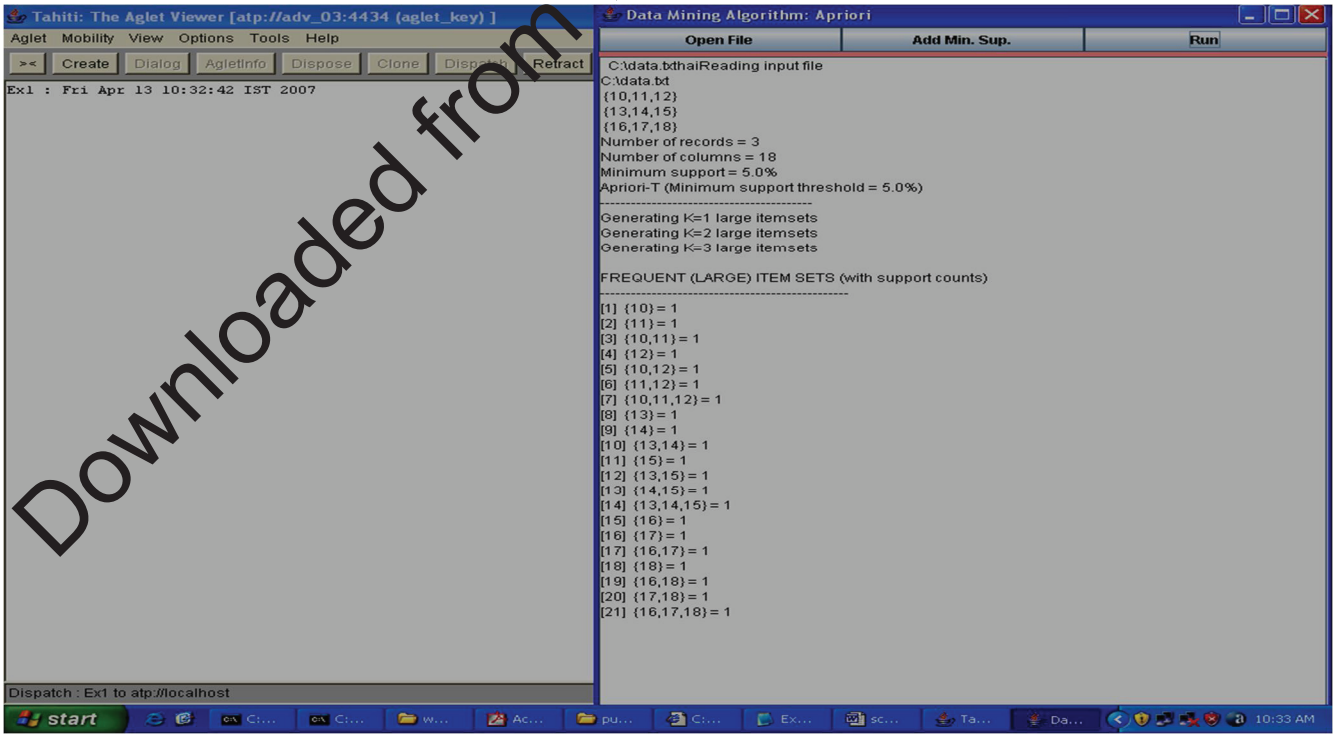

Figure 3: clicking on the run button to get the output

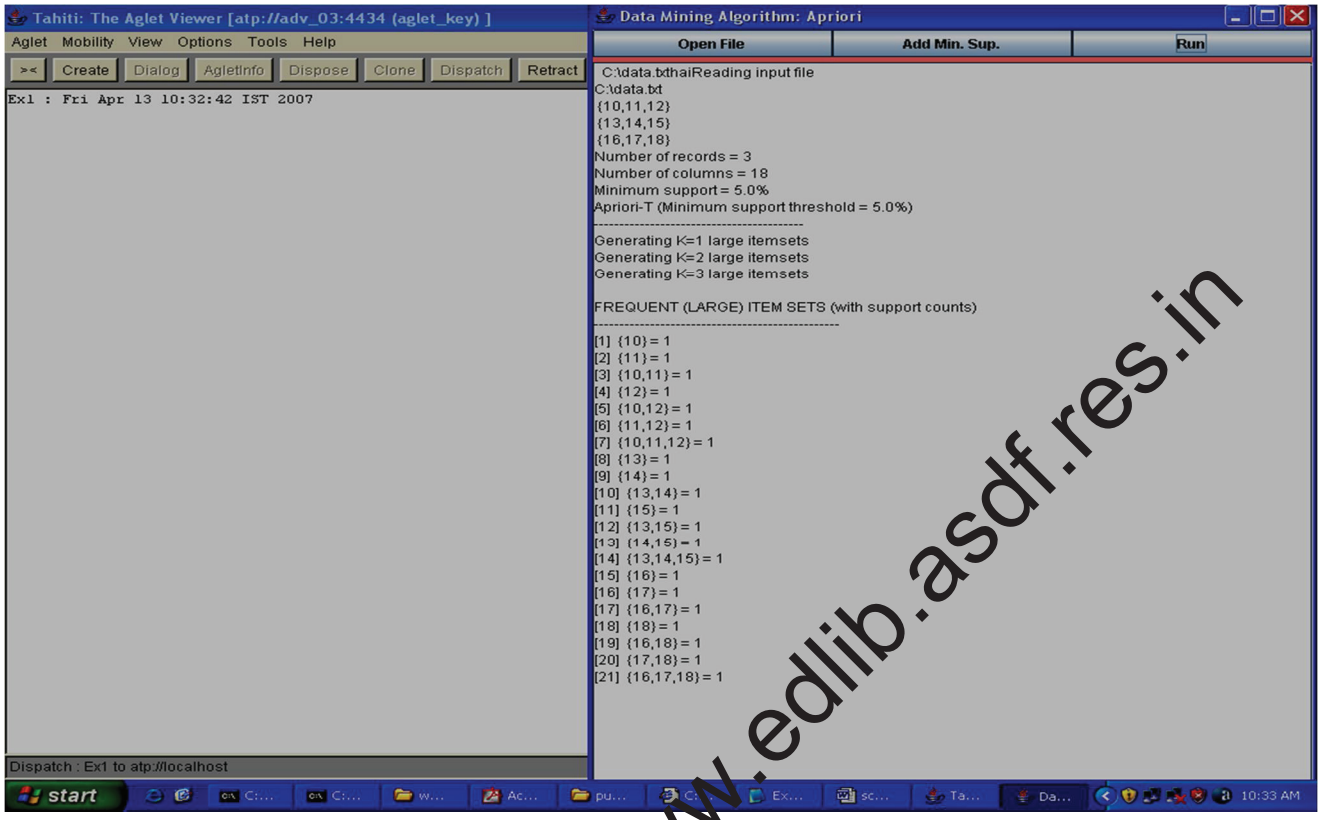

Figure 4: Aglet dispatched to another host

The architecture sends the agent to another server. After dispatching agent, new agent on the current server does no more exists on your server. The protocol for the destination URL is Agent Transfer Protocol (ATP). A dispatched agent from a remote server. First specify the target server and you will get a list of agents on the target server. Then, you can specify one of the aglets form the server. The architecture of distributed data mining will act on data Warehouses located in remote locations. The architecture works on DM with different heterogeneous data. The aglet dispatch to another host is shown in the Figure 4.

VI. Conclusion

This paper proposes the solutions to the issue of knowledge discovery in distributed data mining with less complexity, and tried with mobile agents with less exchange of data distribution, Mobile agents are used for candidate distribution, by conducting the above experiments, proved that mobile agents can be as used for candidate distribution in distributed data mining efficiently. And security of communication agents for reducing processing and storage issues , so some research has to be done in these areas or simplify the use of the mechanisms available like some tools, investigation to be carried out in the area of security design tools , so authors are working in these areas.

Acknowledgment

The authors would like to express their sincere gratitude to the Management of their concerned institution, Hyderabad for their constant encouragement and co-operation.

References

1. R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In VLDB '94: Proceedings of the 20th International Conference on Very Large Data Bases, pages 487--499, San Francisco, CA, USA, 1994. Morgan Kaufmann Publishers Inc.

2. High-performance data mining Parallelized scoring performance with an SAS PMML model in InfoSphere Balanced Warehouse by Jack Baker jjbaker@us.ibm.com), Support Specialist, IBM.
3. Effect of Data Distribution in Parallel Mining of Associations by Agrawal and Srikant, 1994.
4. 2nd International Workshop on Data Mining Standards, Services and Platforms kdd 2004.
5. Agents and Stream Data Mining: A New Perspective May/June 2005 (vol. 20 no. 3)
6. M.J. Zaki, "Scalable Algorithms for Association Mining," IEEE Trans. Knowledge and Data Eng., vol.12 no.2, 2000, pp. 372-390.
7. R. Agrawal and J.C. Shafer, "Parallel Mining of Association Rules," Distributed Systems Online March 2004.
8. 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.
9. A. Prodromidis, P. Chan, and S. Stolfo. Chapter Meta learning in distributed data mining systems: Issues and approaches. AAAI/MIT Press, 2000.
10. S. Cai, Y. Zou, B. Xie, and W. Shao. Mining the web of trust for web services selection. In Web Services, 2008. ICWS '08.IEEE International Conference, pages 809 –810, sept. 2008.
11. S. Chen, Z. Feng, H. Wang, and T. Wang. Building the semantic relations-based web services registry through services mining. In Computer and Information Science, 2009.
12. K. Elgazzar, A. Hassan, and P. Martin. Clustering wsdl documents to bootstrap the discovery of web services. In WebServices (ICWS), 2010 IEEE International Conference, pages 147 –154, July 2010.