INTERNATIONAL JOURNAL OF RESEARCH IN COMPUTER APPLICATIONS AND ROBOTICS
ISSN 2320-7345

INCREASING THE ACCESSIBILITY OF DATA SETS BY USING DISTRIBUTED ALGORITHM IN DATA MINING

C.B.Sivaparthipan[1], S.Raja Ranganathan[2], Prabakar.D[3], Dr.T.Kalaikumaran[4]

[1]*Assistant Professor, E-mail: sivaparthipanece@gmail.com*
[2]*Assistant Professor, E-mail:rajaranganathan.s@gmail.com*
[3]*Assistant Professor, E-mail:prabakaralam@gmail.com*
[4]*Professor & Head, E-mail:proftkalaikumaran@gmail.com*

Assistant Professor- Dept of CSE /SNS Tech Coimbatore, India, 9942033952,.www.snsct.org

Abstract

Association rule mining is an active data mining research area. However, most ARM algorithms cater to a centralized environment. In contrast to previous ARM algorithms, ODAM is a distributed algorithm for geographically distributed data set that reduces communication costs. Modern organizations are geographically distributed. Typically, each site locally stores its ever increasing amount of day-to-day data. Using centralized data mining to discover useful patterns in such organizations' data isn't always feasible because merging data sets from different sites into a centralized site incurs huge network communication costs. Distributed data mining has thus emerged as an active sub area of data mining research. In this project we develop and implement a distributed algorithm, called Optimized Distributed Association Mining, for geographically distributed data sets. ODAM generates support counts of candidate item sets quicker than other DARM algorithms and reduces the size of average transactions, data sets, and message exchanges.

Key Words: Data Sets, Patterns, Message Exchange

1. Introduction

With the rapid increase of stored data in digital form, the interest in the discovery of hidden information has exploded in the last decade. One approximation to the problem of discovery of hidden information is based on finding frequent associations between elements in sets, also called basket analysis. One important special case arises when this approach is applied to the treatment of sequential data. The sequential nature of the problem is relevant when the data to be mined is naturally embedded in a one dimensional space, i.e., when one of the relevant pieces of information can be viewed as one ordered set of elements. This variable can be time or some other dimension, as is common in other areas, like bioinformatics. We define sequential pattern mining as the process of discovering all sub-sequences that appear frequently on a given sequence database and have minimum support threshold. One challenge resides in performing this search in and efficient way.

Data mining software allows users to analyze large databases to solve business decision problems. Data mining is, in some ways, an extension of statistics, with a few artificial intelligence and machine learning twists thrown in. Like

statistics, data mining is not a business solution, it is just a technology. For example, consider a catalog retailer who needs to decide who should receive information about a new product. The information operated on by the data mining process is contained in a historical database of previous interactions with customers and the features associated with the customers, such as age, zip code, and their responses. The data mining software would use this historical information to build a model of customer behavior that could be used to predict which customers would be likely to respond to the new product. By using this information a marketing manager can select only the customers who are most likely to respond. The operational business software can then feed the results of the decision to the appropriate touch point systems (call centers, direct mail, web servers, email systems, etc.) so that the right customers receive the right offers.

We propose a projection-based, sequential pattern-growth approach for efficient mining of sequential patterns. In this approach, a sequence database is recursively projected into a set of smaller projected databases, and sequential patterns are grown in each projected database by exploring only locally frequent fragments. Based on an initial study of the pattern growth-based sequential pattern mining, Free Span, we propose a more efficient method, called ODAM, which offers ordered growth and reduced projected databases. To further improve the performance, a pseudo projection technique is developed in ODAM. A comprehensive performance study shows that ODAM, in most cases, outperforms the apriori-based algorithm.

2. Related Work

Mafruz Zaman Ashrafi, David Taniar, Kate Smith of Monash University published an IEEE computer society on ODAM: An Optimized Distributed Association Rule Mining Algorithm at march 2010.They theoretically proved that association rule mining is an active data mining research area. However, most ARM algorithms cater to a centralized environment. In contrast to previous ARM algorithms, ODAM is a distributed algorithm for geographically distributed data sets that reduces communication costs.

Murat Kantarcýo¡glu and Chris Clifton, Senior Member, IEEE has discussed a paper on Privacy-preserving Distributed Mining of Association Rules on Horizontally Partitioned Data. They stated that data mining can extract important knowledge from large data collections – but sometimes these collections are split among various parties. Privacy concerns may prevent the parties from directly sharing the data, and some types of information about the data. This paper addresses secure mining of association rules over horizontally partitioned data. The methods incorporate cryptographic techniques to minimize the information shared, while adding little overhead to the mining task.

Shenzhi Li, Aditya P. Belapurkar, Christopher D. Janneck Murat Can Ganiz, William M. Pottenger, Tianhao Wu Lehigh University Department of Computer Science and Engineering they have proposed a paper on D-HOTM: Distributed Higher Order Text Mining.It is a framework for Distributed Higher Order Text Mining based on named entities extracted from textual data that are stored in distributed relational databases. Unlike existing algorithms, D-HOTM requires neither full knowledge of the global schema nor that the distribution of data be horizontal or vertical. D-HOTM discovers rules based on higher-order associations between distributed database records containing the extracted entities.A theoretical framework for reasoning about record linkage is provided to support the discovery of higher-order associations. In order to handle errors in record linkage, the traditional evaluation metrics employed in ARM are extended. The implementation of D-HOTM is based on the TMI [29] and tested on a cluster at the National Center for Supercomputing Applications (NCSA).Results on a dataset simulating an important DEA methamphetamine case demonstrate the relevance of D-HOTM in law enforcement and homeland defense.

With the existence of many large transaction databases, the huge amounts of data, the high scalability of distributed systems, and the easy partitioning and distribution of a centralized database, it is important to investigate efficient methods for distributed mining of association rules. The study discloses some interesting relationships between locally large and globally large item sets and proposes an interesting distributed association rule mining algorithm, FDM (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. A performance study shows that FDM has a superior performance over the direct application of a typical sequential algorithm. Further performance enhancement leads to a few variations of the algorithm (D.W. Cheung, et al.2006).

T. Shintani and M. Kitsuregawa as proposed four parallel algorithms (NPA, SPA, HPA and HPA-ELD) for mining association rules on shared nothing parallel machines to improve its performance. In NPA, candidate itemsets are just copied amongst all the processors, which can lead to memory overflow for large transaction databases. The remaining three algorithms partition the candidate itemsets over the processors. If it is partitioned simply (SPA), transaction data has to be broadcast to all processors. HPA partitions the candidate itemsets using a hash function to eliminate broadcasting, which also reduces the comparison workload significantly. HPA-ELD fully utilizes the available memory space by detecting the extremely large itemsets and copying them, which is also very effective at flattering the load over the processors. We implemented these algorithms in a shared nothing environment. Performance evaluations show that the best algorithm, HPA-ELD, attains good linearity on speedup ratio and is effective for handling skew (2006).

M.J. Zaki surveys the state of the art in parallel and distributed association-rule-mining algorithms and uncovers the field's challenges and open research problems. This survey can serve as a reference for both researchers and practitioners (Oct.-Dec. 2009).Association rule discovery has emerged as an important problem in knowledge discovery and data mining. The association mining task consists of identifying the frequent itemsets and then, forming conditional implication rules among them. In this paper, we present efficient algorithms for the discovery of frequent itemsets which forms the compute intensive phase of the task. The algorithms utilize the structural properties of frequent itemsets to facilitate fast discovery. The items are organized into a subset lattice search space, which is decomposed into small independent chunks or sublattices, which can be solved in memory. Efficient lattice traversal techniques are presented which quickly identify all the long frequent itemsets and their subsets if required. We also present the effect of using different database layout schemes combined with the proposed decomposition and traversal techniques. We experimentally compare the new algorithms against the previous approaches, obtaining improvements of more than an order of magnitude for our test databases (M.J. Zaki, 2011).The problem of online mining of association rules in a large database of sales transactions. The online mining is performed by preprocessing the data effectively in order to make it suitable for repeated online queries. We store the preprocessed data in such a way that online processing may be done by applying a graph theoretic search algorithm whose complexity is proportional to the size of the output. The result is an online algorithm which is independent of the size of the transactional data and the size of the preprocessed data. The algorithm is almost instantaneous in the size of the output. The algorithm also supports techniques for quickly discovering association rules from large itemsets. The algorithm is capable of finding rules with specific items in the antecedent or consequent. These association rules are presented in a compact form, eliminating redundancy. The use of nonredundant association rules helps significantly in the reduction of irrelevant noise in the data mining process this concept is proposed by C.C. Aggarwal and P.S. Yu (2011).

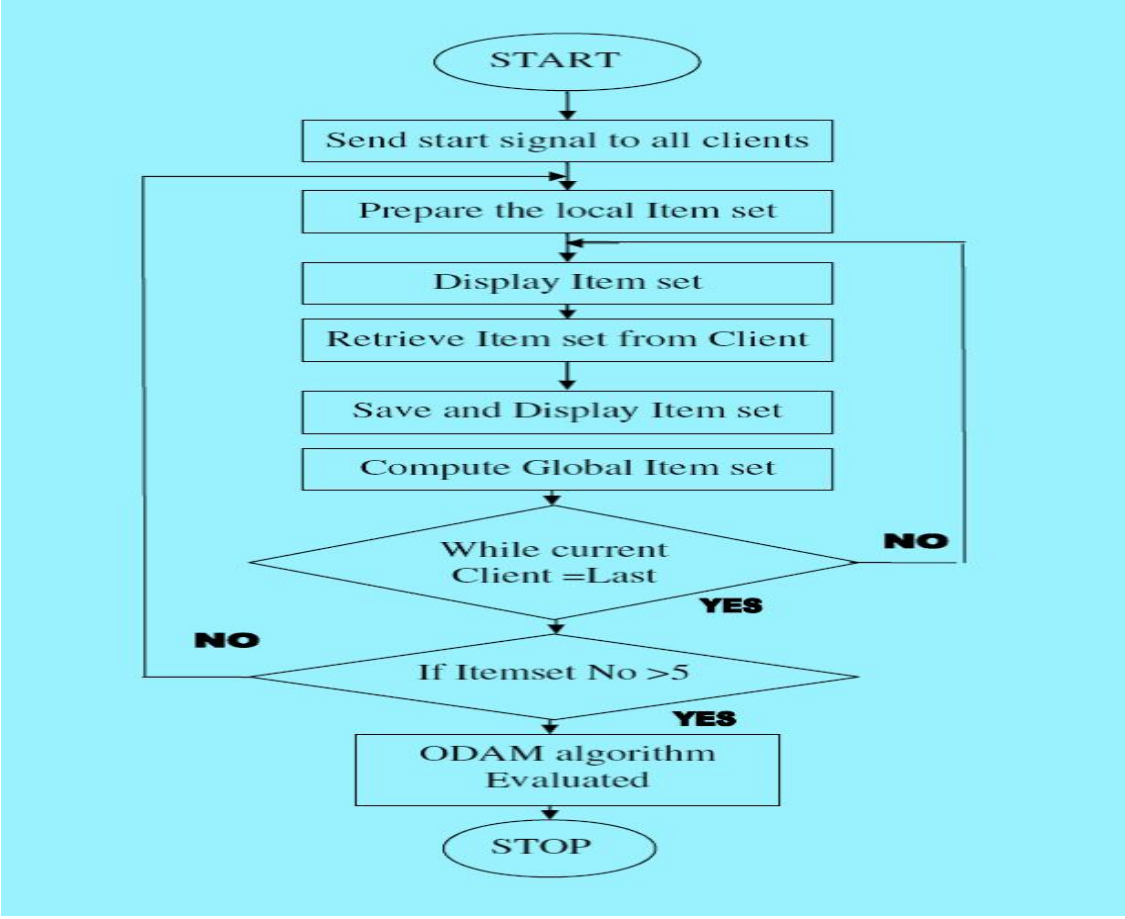
R. Aggarwal and J.C. Shafer consider the problem of mining association rules on a shared-nothing multiprocessor. We present three algorithms that explore a spectrum of trade-offs between computation, communication, memory usage, synchronization, and the use of problem-specific information. The best algorithm exhibits near perfect scaleup behavior, yet requires only minimal overhead compared to the current best serial algorithm (March 2012).Many parallel or distributed ARM algorithms exist in the data mining literature. However, most were designed for shared memory parallel environments. Based on the nature and implementation of each algorithm, we can divide the existing algorithms into two groups: parallel ARM and DARM. We can categorize parallel ARM algorithms as data-parallelism or task-parallelism algorithms. In the former, the algorithms partition the data sets among different nodes; in the latter, each site performs the task independently but must access the entire data set. Count Distribution (CD) algorithm, Data Distribution is a task-parallelism-based algorithm, PEAR algorithm are few parallel ARM algorithms.

To overcome these problems, we don't generate candidate support counts from the raw data set after the first pass. This is because item sets that are infrequent in the first pass cannot generate frequent item sets in a subsequent pass. To efficiently generate candidate support counts of later passes, ODAM eliminates all infrequent items after the first pass and places those new transactions into the main memory. This technique not only reduces the average transaction length but also reduces the 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. Moreover, the number of infrequent item sets increases proportionally for higher support thresholds.

3. Main Idea

ODAM provides an efficient method for generating association rules from different datasets, distributed among various sites. In future work, we plan to investigate how to efficiently perform DARM on different organizations in different domains. Because security and privacy is a common issue for data mining application, we'll also investigate how to maintain DARM's privacy without increasing overall communication costs.It will be enhanced two cutting edge technologies agents and data mining. By integrating these two technologies, the power for each of them is enhanced. Integrating agents into data mining systems, or constructing data mining systems from agent perspectives, the flexibility of data mining systems can be greatly improved. New data mining techniques can add to the systems dynamically in the form of agents, while the out-of-date ones can also be deleted from systems at run-time. Equipping agents with data mining capabilities, the agents are much smarter and more adaptable. In this way, the performance of these agent systems can be improved. A new way to integrate these two techniques –ontology-based integration will be decided.

We have extensively studied ODAM's performance to confirm its effectiveness. We implemented ODAM using VB.NET. We established a socket-based, client-server distributed environment to evaluate ODAM's message reduction techniques. Each site has a receiving and a sending unit and assigns a specific port to send and receive candidate support counts. Because the candidate item sets that each site generates will be based on the global frequent item set for the previous pass, the candidate item sets are Identical among various sites. We chose two real data sets for this implementation. Characteristics of each data set, including the number of items, average transaction size, and number of transactions of each data set illustrated in Figure 1. Data sets are developed by us and it can be extended for any number of transactions.



3.1 ODAM–Algorithm

ODAM first computes support counts of 1-itemsets from each site in the same manner as it does for the sequential Apriori. It then broadcasts those item sets to other sites and discovers the global frequent 1-itemsets. Subsequently, each site generates candidate 2-itemsets and computes their support counts. At the same time, ODAM also eliminates all globally infrequent 1-itemsets from every transaction and inserts the new transaction (that is, a transaction without infrequent 1-itemset) into memory. While inserting the new transaction, it checks whether that transaction is already in the memory. If it is, ODAM increases that transaction's counter by one. Otherwise, it inserts the transaction with a count equal to one into the main memory. After generating support counts of candidate 2-itemsets at each site, ODAM generates the globally frequent 2-itemsets.

It then iterates through the main memory (transactions without infrequent 1-itemsets) and generates the support counts of candidate item sets of respective length. Next, it generates the globally frequent item sets of that respective length by broadcasting the support counts of candidate item sets after every pass. ODAM algorithm as given in Figure.2.

```
NF = {Non-frequent global 1-itemset}
For all transaction t € D
{
For all 2-subsets s of t
If (s € c2) s.sup ++;
T ' =delete_ nonfrequent_ items (t);
Table.add (t ');
}
Send_to_receiver (C2) ;
F2 = receive_from_receiver (Fg ) ;
C3 = {Candidate itemset};
T=Table.getTransactions ( ); k =3;
While (Ck ≠ { }) {
For all transaction t € T
For all k-subsets s of t
if ( s € cK ) s.sup ++ ;
K ++;
Send_to_receiver (Ck ) ;
Ck + 1 = {Candidate itemset};
}
```

Figure 2 : ODAM Algorithm

3.2 Established a socket-based, client-server distributed environment

We establish a socket-based, client-server distributed environment to evaluate ODAM's message reduction techniques. Each site has a receiving and a sending unit and assigns a specific port to send and receive candidate support counts. Because the candidate item sets that each site generates will be based on the global frequent item set for the previous pass, the candidate item sets are identical among various sites.

3.3. ODAM -An Algorithm Implementation

This module consists developing steps involved in ODAM algorithm. Frequent 1-itemsets from each site in the same manner as it does for the sequential Apriori. It then broadcasts those item sets to other sites and discovers the global frequent 1-itemsets.Subsequently, each site generates candidate 2-itemsets and computes their support counts. At the same time, ODAM also eliminates all globally infrequent 1-itemsets from every transaction and inserts the new transaction (that is, a transaction without infrequent 1-itemset) into memory. While inserting the new transaction, it checks whether that transaction is already in the memory. If it is, ODAM increases that transaction's counter by one.

3.4 Frequent item sets generation

This module shows how ODAM can efficiently generate support counts; we conduct an experiment in a single site with different support, shows ODAM total execution time for generating the frequent item sets of various lengths using the data sets. Generating support counts of candidate item sets for each iteration takes approximately three times longer than it takes for the previous iteration. ODAM removes a significant number of infrequent 1-itemsets from every transaction after the first pass, so it finds a significant number of identical transactions. After eliminating infrequent items, ODAM doesn't enumerate candidate item sets multiple times for any identical transaction. Furthermore, it requires a minimal number of comparison and update operations to generate support because it doesn't require comparison and update operations multiple times for similar transactions.

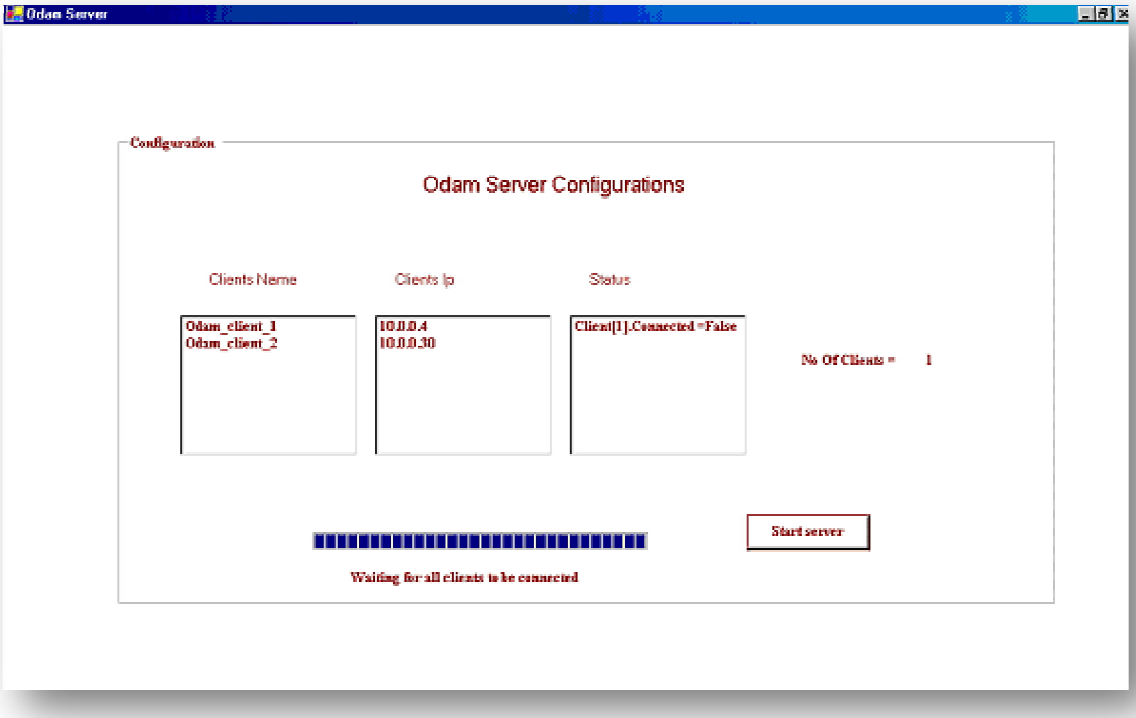


Figure 3: Connection made between clients

3.5 Message exchange optimization

To reduce communication costs, we highlight message optimization technique. We can divide the message optimization techniques into two methods direct and indirect support counts exchange. Each method has different aims, expectations, advantages, and disadvantages. For example, the first method exchanges each candidate item sets support count to generate globally frequent item sets of that pass. All sites share a common globally frequent item set with identical support counts, so rules that are generated at different participating sites have identical confidence. This approach focuses on a rule's exactness and correctness.

4. Conclusion

ODAM provides an efficient method for generating association rules from different datasets, distributed among various sites. In future work, we plan to investigate how to efficiently perform DARM on different organizations in different domains. Because security and privacy is a common issue for data mining application, we'll also

investigate how to maintain DARM's privacy without increasing overall communication costs. It will be enhanced two cutting edge technologies agents and data mining. By integrating these two technologies, the power for each of them is enhanced. Integrating agents into data mining systems, or constructing data mining systems from agent perspectives, the flexibility of data mining systems can be greatly improved. New data mining techniques can add to the systems dynamically in the form of agents, while the out-of-date ones can also be deleted from systems at run-time. Equipping agents with data mining capabilities, the agents are much smarter and more adaptable. In this way, the performance of these agent systems can be improved. A new way to integrate these two techniques –ontology-based integration will be decided.

REFERENCES

1. R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules in Large Database,"Proc. 28th Int'l Conf. Very Large Databases (VLDB 94), Morgan Kaufmann, 2007, pp. 407-419.

2. R. Agrawal and J.C. Shafer, "Parallel Mining of Association Rules,"IEEE Tran. Knowledge and 16 IEEE Distributed Systems Online March2004DataEng.vol.8,no.6,2006,pp.962-969; http://csdl.computer.org/comp/trans/tk/1996/06/k0962abs.htm.

3. 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, 2011,pp. 527-540;
http://csdl.computer.org/comp/trans/tk/2001/04/k0527abs.htm.

4. C.L. Blake and C.J. Merz , UCI Repository of Machine Learning Databases, Dept. of Information and Computer Science, University of California, Irvine, 2008;
www.ics.uci.edu/~mlearn/MLRepository.html.

5. D.W. Cheung, et al., "A Fast Distributed Algorithm for Mining Association Rules," Proc.Parallel and Distributed Information Systems, IEEE CS Press, 2006, pp. 31-42;
http://csdl.computer.org/comp/proceedings/pdis/1996/7475/00/74750031abs.htm.

6. 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.911922;
http://csdl.computer.org/comp/trans/tk/2006/06/k0911abs.htm.

7. J. Han, J. Pei, and Y. Yin, "Mining Frequent Patterns without Candidate Generation,"Proc. ACM SIGMOD Int'l. Conf. Management of Data, ACM Press, 2011, pp. 1-12.

8. J.S. Park, M. Chen, and P.S. Yu, "An Effective Hash Based Algorithm for Mining Association Rules,"Proc. 2005 ACM SIGMOD Int'l Conf. Management of Data, ACM Press, 2005, pp. 175-186.

9. A. Savasere, E. Omiecinski, and S.B. Navathe, "An Efficient Algorithm for Mining Association Rules in Large Databases,"Proc. 29st Int'l Conf. Very Large Databases (VLDB 10), Morgan Kaufmann, 2010, pp. 432-444.

10. A. Schuster and R. Wolff, "Communication-Efficient Distributed Mining of Association Rules," Proc. ACM SIGMOD Int'l Conf. Management of Data, ACM Press, 2011,pp. 473-484.

11. 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;
http://csdl.computer.org/comp/proceedings/pdis/2006/7475/00/74750019abs.htm.

12. G.W. Webb, "Efficient Search for Association Rules,"Proc. Sixth ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD 00), ACM Press, 2011, pp. 99-

13. M.J. Zaki and Y. Pin, "Introduction: Recent Developments in Parallel and Distributed Data Mining,"J. Distributed and Parallel Databases, vol. 11, no. 2, 2012, pp. 123-127.

14. M.J. Zaki , "Scalable Algorithms for Association Mining,"IEEE Trans. Knowledge and Data Eng. , vol. 12 no. 2, 2000,pp. 372-390; http://csdl.computer.org/comp/trans/tk/2012/03/k0372abs.htm.

15. 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, 2008.

16. M.J. Zaki, "Parallel and Distributed Association Mining: A Survey,"IEEE Concurrency, Oct.-Dec. 2009, pp. 14-25; http://csdl.computer.org/comp/mags/pd/1999/04/p4014abs.htm.

Biography

C.B.Sivaparthipan, received B.E. degree in Computer Science and Engineering from VLBJCET-Coimbatore, India and M.Tech. degree in Computer Science and Engineering from Hindustan university– Chennai, in 2010 and 2012.He is now working as an Assistant Professor at SNS College of Technology, Coimbatore – India and his area of interest mainly focusing on Mobile Ad-Hoc Network with Data Mining Application.

S.Raja Ranganathan, received MCA degree in Computer Science Applications from Anna University-Chennai, India and M.E. degree in Computer Science and Engineering from Anna university of Technology – Coimbatore, India, in 2008 and 2010.At present, He is an Assistant Professor of Computer Science and Engineering in SNS College of Technology, Coimbatore. His research interest focuses on Distributed Data Mining ,Web Semantic and Server side Applications.

Prabakar.D, received B.E. degree in Computer Science and Engineering from Anna University-Chennai, India and M.E. degree in Computer Science and Engineering from Anna university of Technology – Coimbatore, India, in 2008 and 2010. At present, He is an Assistant Professor of Computer Science and Engineering in SNS College of Technology, Coimbatore. His research interest focuses on Wireless Communication, Mobile Computing and Wireless Sensor Networks.

Dr.T.Kalaikumaran, is presently Professor & Head, Department of Computer Science & Engineering, SNS College of Technology, affiliated to Anna University- Chennai, Tamilnadu, India. M.E Degree and Ph.D Degree from Anna University Chennai. His research interests include network security, web services and data mining. Further more he has been published many papers in international journal and conference papers. He is an active member of IEEE and Indian Computer Society.