ijecs open access www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 6 Issue 6 June 2017, Page No. 21713-21717
Index Copernicus value (2015): 58.10 DOI: 10.18535/ijecs/v6i6.26

Flexible Approach for Data Mining using Grid based Computing Concepts

Abdul Ahad[1], Dr.Y.Suresh Babu[2]
[1] Department of CSE, NITS, Hyderabad, Telangana, INDIA
[2] Department of CSE, JKC, Guntur, Andhra Pradesh, INDIA
*1 ahadbabu@gmail.com; 2 yalavarthi_s@yahoo.com;

Abstract. Now days, in the field of life sciences and business, knowledge discovery has become a common task in both for the growing amount of data being gathered and for the complexity of the analysis that need to be performed on it. Due to some unique characteristics of today's data sources, such as their heterogeneity, high dimensionality, distributed nature and large volume. Distribution of data and computation allows increasing trend towards decentralized business organizations; distribution of users, software, and hardware systems magnifies the need for more advanced and flexible approaches and solutions. Here we present the state of the art about the major data mining techniques, systems and approaches. This paper discusses how distributed and Grid computing can be used to support distributed data mining. In particular, a distinction is made between distributed and Grid-based data mining methods.

Keywords: Data mining, Distributed data, Grid computing, Knowledge discovery, Data sharing.

1 Introduction

The main objective of data mining is to extract the hidden information from large scale data repositories like databases and digital libraries. And it's also used for building valuable knowledge patterns and predictive models. The data mining main tasks are classification, clustering and association rules discovery. Data mining is a complex computing task that deals with memory resident data. The major difference between distributed data mining and the grid data mining are the number of participated computing nodes, the degree of the data distribution and the costs of the communication. The distributed data mining deals with the loosely coupled systems such as slow networks. The Grid is a distributed computing infrastructure that enables coordinated resource sharing within dynamic organizations consisting of individuals, institutions, and resources. The aim of grid computing is to give organizations and application developers the ability to create distributed computing environments that can utilize computing resources on demand. The Grid computing helps user to achieve much faster results on large operations and at lower costs. The advanced technologies in network have produced large amount of data stored on purely distributed databases and repositories. The grid data mining shares many common features with parallel data mining and distributed data mining, there are stage peculiarities and requirements pretend that efforts and get results in such region cannot be compared with those achieved in parallel data mining and disturbed data mining.

2 The Distributed Data Mining Techniques

Through the growth of related data, databases in today's world are highly distributed. Most of the organizations which have geographical separation does mining on distributed data sources and collects the data which is consistent and well formed for their characterized. The cost of retrieval/storage becomes more un-affordable as the data is inherently distributed across the sources. Distributed data mining is the novel approach arrived due to the consequences of data mining and inherited distribution of data.

There are two possibilities in distributed sources, namely homogeneous and heterogeneous data sources. Single global source is the only source through which homogeneity and heterogeneity are possible, where horizontal partitioned collection of data leads to earlier and vertical partitioned collection of data leads to later. Matching process is from the tuple selection that is identified uniquely across the data sources.

Table 1. Distributed data mining techniques

	Paradigm	Platform	Communication cost
Association rules	Data parallelism	Share-nothing	Exponential
Classification	Data parallelism	Share-nothing	Linear
Clustering	Data parallelism	Share-nothing	Linear

Clients, data source, supporting hardware and mining software formed with artificial intelligence methodologies are the components of data mining environment. In distributed data mining, it resolves issues like distribution of clients, data sources, supporting hardware and the mining aspects. Most of the people believe that distributed data mining deals with the distribution of subsets of data either physically or geographically.

2.2. Organization of distributed association rules, classification, and clustering algorithms

The majority of distributed association-rules algorithms are directly derived from their parallel versions. This kind of task needs high volumes of communication at each step: this makes it not scalable on distributed environments where the network speed is normally low. On the other hand, distributed classification approaches are based on ensemble methods. These approaches minimize considerably the communication cost but decrease the predictive performance and yield reduced efficiency. The most important phase in a classification or clustering algorithm is the aggregation of local models. This phase is decisive for the quality of the final model. This is shown as below (**Fig. 1.**):

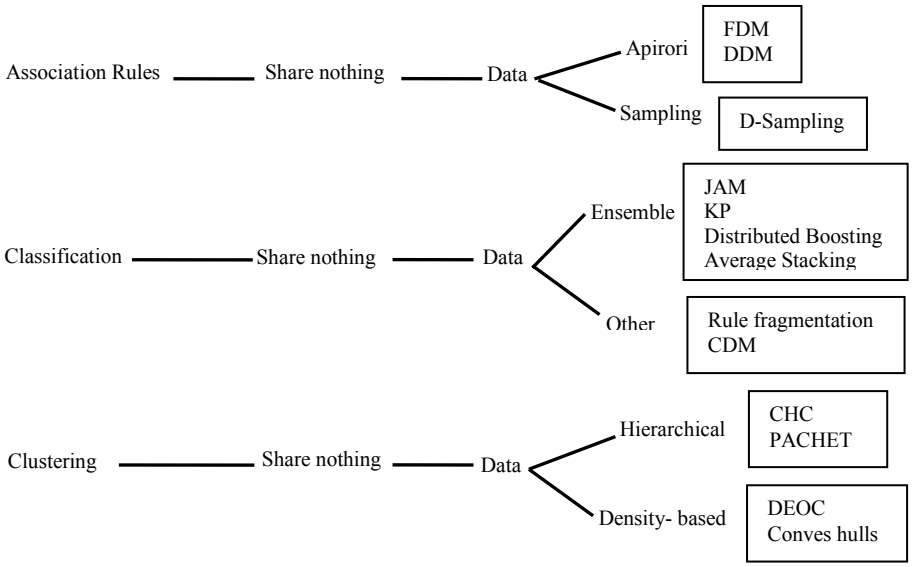

Fig. 1. Organization of Distributed Association rule, Classification, and clustering algorithms

2.3. Algorithms used for distributed data mining:

Count Distribution: To achieve parallelism this algorithm partitions the data source into manageable data units. Total database is distributed among N workstations as $(1/N)^{th}$ of the database. Like Apriori algorithm, the frequent subsets are extended one at a time. However, it differs from Apriori in the communication phase, where all workstations exchange the data partitions which are frequently occurred.

Data Distribution: For each work station memory bandwidth need to be increased and computational redundancy need to be decreased as per this algorithm design. Maximum frequency item set candidates are partitioned by this algorithm.

Candidate Distribution: Selective partitioning of candidates over workstations is the major approach in this algorithm through which it reduces storage/retrieval costs. This algorithm follows optimal number of fixed passes to achieve selective partitioning.

2.4. Software tools for distributed data mining:

The RapidMiner (formerly YALE) Distributed Data Mining Plug-in allows performing distributed data mining experiments in a simple and flexible way. The experiments are not actually executed on distributed network nodes. The plug-in only simulate this. Simulation makes it easy to experiment with diverse network structures and communication patterns.

The service oriented architecture (SOA) paradigm can be exploited for the implementation of data and knowledge-based applications in distributed environments. The Web Services Resource Framework (WSRF) can be exploited for developing high-level services for distributed data mining applications. Weka4WS adopts the WSRF technology for running remote data mining algorithms and managing distributed computations. The Weka4WS user interface supports the execution of both local and remote data mining tasks.

3.1. The implementation of Data mining Grid

Most grid classifiers have their foundations in ensemble way and it has been applied in various domains to increase the classification accuracy of predictive models. It produces multiple models (base classifiers) – typically from "homogeneous" data subsets – and combines them to enhance accuracy. In this approach, supervised learning techniques are first used to learn

classifiers at local data sites; then meta-level classifiers are learned from a data set generated using the locally learned concepts. Meta-learning follows three main steps:

i. Concrete base classifiers at each site using a classifier learning algorithms.
ii. Collect the base classifiers at a central site. Produce meta-level data from a separate validation set and predictions generated by the base classifier on it.
iii. Generate the final classifier (meta-classifier) from meta-level data grid.

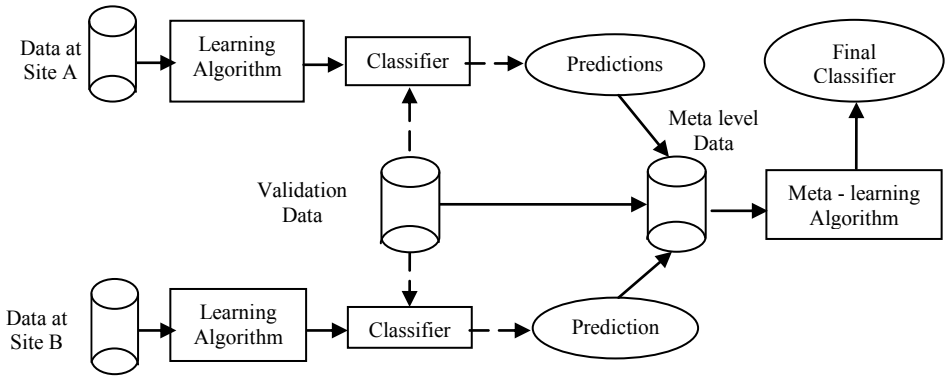

Fig. 2. Grid Meta Learning from Distributed Data Sites

3.2. Algorithms used for Grid environment

The Grid-based Distributed Max-Miner (GridDMM) (Luo, 2006) is an algorithm for mining maximal frequent itemsets from databases on a data Grid system. To deal with Grid environment, it is designed to have low communication and synchronization overhead. GridDMM consists of a local mining phase and a global mining phase. During the local mining phase, each node mines the local database to discover the local maximal frequent itemsets. In the global mining phase they form a set of maximal candidate itemsets for a top-down search.

The DisDaMin project (DIStributed DAta MINing) (Fiolet, 2006) proposes new exploitable algorithms for Grids. First DisDaMin fragments the data using clustering methods then uses asynchronous collaborative techniques according to the specificities of execution on Grids. Simulations are performed on the French national Grid GRID5000 (built on top of Xtremweb middleware) showing the efficiency of DisDAMin. A method of integrating the Apriori algorithm in distributed databases with the Globus Toolkit is proposed in Aflori (2007).

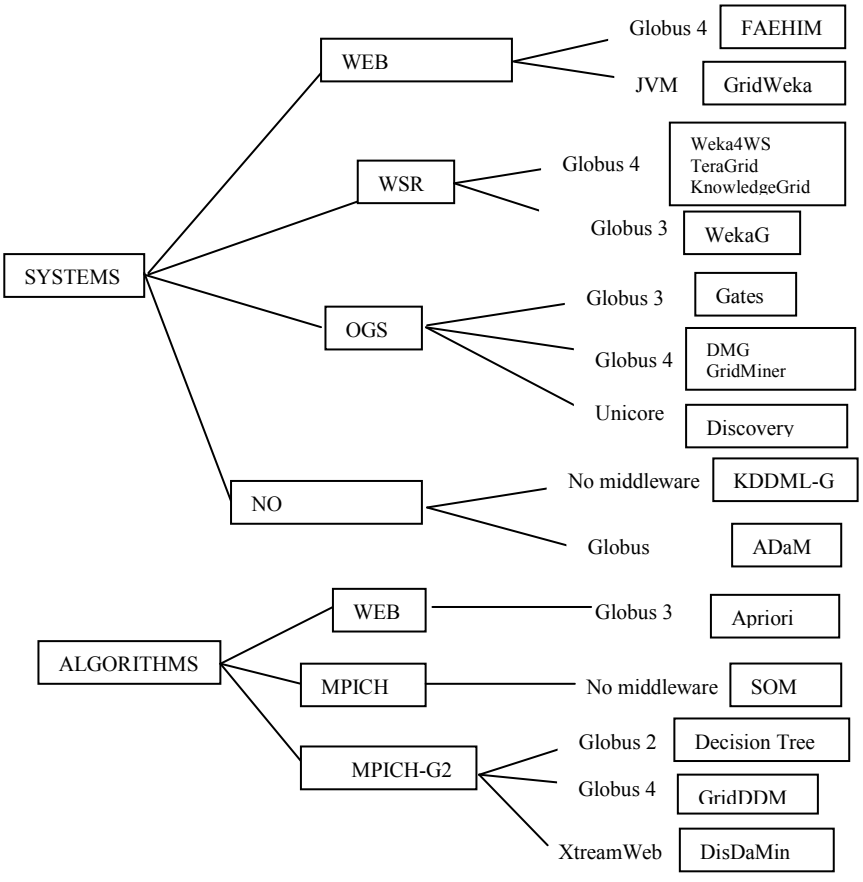

Fig. 3. Architecture for decision tree model in the Grid environment

Teragrid is the largest storage and computing infrastructure including managed services to build scientific applications. But it only offers the possibility to use specific tools, it is not possible not to deploy or integrate custom solutions. By contrast, Infogrid provides an integration mechanism based on Wrappers. Infogrid is more flexible, it permits to the user to have individualized view of the resources and dynamic data integration by providing a universal query system and a wrapper interface. Teragrid uses Globus 4 as grid-middleware, OGSA-DAI for data management and implements WSRF-compliant services, but Infogrid mentions only the use of OGSA-DAI.

The benefits of Infogrid are proved when it is adopted by the Discovery Net project. This framework supports data integration functionalities provided by Infogrid wrappers. ADaM has its proper composition mechanism, Grid Miner provides a workflow engine for sequential or parallel executions, GATES uses a pipeline model, the Knowledge Grid provides a graphical interface (VEGA) for designing complex applications, DataMiningGrid is based on the Triana engine, and KDDML-G provides a graphical interface for complex query composition.

The WSRF standard is supported by the Knowledge Grid and DataMiningGrid systems where Discovery Net, Grid Miner, and GATES are based on OGSA architecture. But ADaM and KDDML-G architecture do not respect any Grid standard. Most of the reviewed systems are based on the Globus middleware. ADaM, Grid Miner, the Knowledge Grid, and DataMiningGrid are based on Globus 4 but GATES is based on Globus 3. Discovery Net uses Unicore and KDDML-G was experimented on a Beowulf cluster. ADaM can be considered as a toolkit rather than a framework. GridWeka and WekaG tried to adopt Weka to a Grid infrastructure using the client server architecture.

4. Conclusions

Data mining techniques are key essentials for knowledge discovery applications aimed at extracting important knowledge, implementing business cleverness strategies, and get better company competitiveness. In this part a considerable set of such techniques has been presented, in particular those addressing the utilization of large and remotely dispersed datasets and/or high-performance computers. Comparisons and evaluations on the presented techniques and approaches have been discussed throughout the different sections of the chapter. The availability of new computing platforms and paradigms has for sure driven the majority of the novelties registered in the last years; in particular Grid computing has represented the most demanding, challenging, and promising of such platforms.

References

1. Andrei L. Turinsky, Robert L. Grossman y "A Framework for Finding Distributed Data Mining Strategies That are Intermediate Between Centralized Strategies and In-Place Strategies", 2004.
2. Assaf Schuster, Ran Wolff, and Dan Trock, "A High-Performance Distributed Algorithm for Mining Association Rules". In Third IEEE International Conference on Data Mining, Florida , USA, November 2003.
3. R. Agrawal and J. C. Shafer, "Parallel Mining of Association Rules". IEEE Transactions On Knowledge And Data Engineering, 8:962-969, 1996.
4. Felicity George, Arno Knobbe, "A Parallel Data Mining Architecture for Massive Data Sets", High Performance Research Center, 2001.
5. Abraham, A., & Nath, B. (2000). Hybrid heuristics for optimal design of artificial neural networks. In R. John & R. Birkenhead (Eds.), Advances in Soft Computing Techniques and Applications (pp. 15-22). Springer-Verlag.
6. Abraham, A., Grosan, C., & Ramos, V. (Eds.). (2006). Swarm Intelligence in Data Mining, Studies in Computational Intelligence. Springer-Verlag.
7. Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In Proceedings of the fifth annual workshop on Computational learning theory (pp. 144-152). ACM.
8. Brezany, P., Hofer, J., Tjoa, A., & Wohrer, A. (2003). Gridminer: An infrastructure for data mining on computational grids. In Data Mining on Computational Grids APAC'03 .
9. Brin, S., Motwani, R., Ullman, J. D., & Tsur, S. (1997). Dynamic itemset counting and implication rules for market basket data. In J. Peckham (Ed.), International Conference on Management of Data (pp. 255-264). ACM Press.
10. Cannataro, M. & Talia, D. (2003). The knowledge grid: An architecture for distributed knowledge discovery. Commun. ACM, 46(1), 89-93.
11. Congiusta, A., Talia, D., & Trunfio, P. (2007). Distributed data mining services leveraging wsrf. Future Generation Computing Systems, 23(1), 34-41.
12. Dhillon, I. S., & Modha, D. S. (2000). A data-clustering algorithm on distributed memory multiprocessors. In Large-Scale Parallel Data Mining (pp. 245-260). Lecture Notes in Artificial Intelligence.
13. Dietterich, T. G. (2000). An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. Machine Learning, 40, 139-157.
14. Freund, Y. & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computing System Science, 55(1), 119-139.
15. Giannadakis, N., Rowe, A., Ghanem, M., & Guo, Y. (2003). Infogrid: providing information integration for knowledge discovery. Information Sciences 155, 199-226.

16. Hall, L. O., Chawla, N., & Bowyer, K. W. (1998). Combining decision trees learned in parallel.
17. Lazarevic, A., & Obradovic, Z. (2002). Boosting algorithms for parallel and distributed learning. Distributed and Parallel Databases, 11(2), 203-229.
18. Lazarevic, A., Pokrajac, D., & Obradovic, Z. (2000). Distributed clustering and local regression for knowledge discovery in multiple spatial databases. In 8th European Symposium on Artificial Neural Networks (pp. 129-134).
19. Lippmann, R. P. (1987). An introduction to computing with neural nets. IEEE ASSP Magazine,4(22), 7-25.
20. Luengo, F., Cofino, A. S., & Gutierrez, J. M. (2004). Grid oriented implementation of self-organizing maps for data mining in meteorology. Lecture Notes in Computer Science, 2970, 163-171.
21. Luo, C., Pereira, A. L., & Chung, S. M. (2006). Distributed mining of maximal frequent itemsets on a data grid system. Journal of Supercomputing, 37(1), 71-90.
22. Romei, A., Ruggieri, S., & Turini, F. (2006). Kddml: a middleware language and system for knowledge discovery in databases. Data Knowledge Engineering, 57(2), 179-220.
23. Romei, A., Sciolla, M., Turini, F., & Valentini, M. (2007). Kddml-g: a grid-enabled knowledge discovery system. Concurr. Comput. : Pract. Exper., 19(13), 1785-1809.
24. Rushing, J., Ramachandran, R., Nair, U., Graves, S., Welch, R., & Lin, H. (2005). ADaM: a data mining toolkit for scientists and engineers. Computers and Geosciences, 31, 607-618.
25. Samatova, N. F., Ostrouchov, G., Geist, A., & Melechko, A. (2002). RACHET: An Efficient Cover-Based Merging of Clustering Hierarchies from Distributed Datasets. Distributed and Parallel Databases, 11(2), 157-180.