

A new Distributed Data Mining system on Grid

Huaiguo Fu, M-Tahar Kechadi

The School of Computer Science and Informatics, University College Dublin, Belfield,
Dublin 4, Ireland.

{huaiguo.fu, tahar.kechadi}@ucd.ie

Abstract. As large amounts of data, which are being generated, collected and stored in many fields of research and applications, continue to grow inexorably in size and complexity, Distributed Data Mining (DDM) is becoming a more and more crucial area for research and applications. Many techniques have been proposed for DDM. However, challenges of DDM such as heterogeneous data, complex data, security, privacy and autonomy of local databases, network topology and transmission scheme, still bother us. We need to develop more scalable and more efficient techniques and systems for DDM. This paper offers a brief overview and discussion of DDM issue and techniques, and presents a new DDM system on grid: ADMIRE, which combines efficient DDM techniques and effective infrastructures on real-world large applications.

1 Introduction

The techniques of data mining are widely used in research and application to look for relationships and knowledge that are implicit in large volumes of data and are interesting in the sense of impacting an organization's practice. For example, data mining techniques can help companies to provide better, customized services and support decision making. However, there are many challenges of data mining posed by very large and complex data sets. One of the main challenges in data mining is the development of efficient techniques that scale up to large and possibly physically distributed data sets. Distributed Data Mining (DDM) is one solution of this challenge. As large amounts of data, which are being generated, collected and stored in many fields of research and applications, continue to grow inexorably in size and complexity, DDM is becoming a more and more crucial area for research and application.

Data mining and knowledge discovery can benefit from the use of DDM techniques to improve mining performance of huge data or distributed data. Although there are many efficient algorithms and techniques for mining centralized data sets, it's inefficient or incapable to deal with huge data sets or distributed data sets.

There are two main reasons to choose DDM. The first one is that data is very large. If data is too large, it's hard to store it at a single site, or it's inefficient or incapable to mine such large data at a single site. In such a case, data may be decomposed into some parts that are distributed at different sites. Then we

perform the data mining operations for each site. At the end, the mining results of each site are combined to gain global results. This will optimize centralized data mining since the work load is distributed among the sites.

The second reason is that we need to deal with inherent distributed data sets. In fact, various wired and wireless networks such as internet, intranets, local area networks, ad hoc wireless networks and sensor networks etc. produce many distributed resources of data. These distributed data need to be mined to gain global patterns, models or knowledge. The straightforward solution is to transfer all data to a central site, where data mining is done. However, even if we have enough capacity to handle the data storage and data mining at a central site, it may be too expensive to transfer the local data sets to the central site. On the other hand, the privacy issue is playing an important role in the emerging distributed data. The distributed data sets may not be transferred because of privacy, security or autonomy of the data sets. Therefore, DDM is an effective and scalable solution for mining huge and distributed data sets in distributed computing environments.

In recent years, DDM has attracted a lot of attention among the fields of research and applications. Many techniques and systems of DDM have been proposed. However, the DDM problems such as heterogeneous data, complex data, security, privacy and autonomy of local databases, network topology and transmission scheme, still bother us. We need to develop more scalable and more efficient techniques for DDM.

We propose a new DDM system on Grid: ADMIRE. Combining dynamic efficient DDM techniques and Grid-based service system, ADMIRE defines a new DDM infrastructure to deal with large and complex real distributed data. The purpose of this system is to develop efficient DDM algorithms that scale up large distributed data sets, integrate efficient DDM algorithms and techniques, and Grid-based distributed computing environment for extracting knowledge from large amounts of large and complex real distributed data for research and commercial application. The emphases of ADMIRE are easy, efficient and flexible, which easy means the system is easy to use, easy to extend, efficient means the system can mine the knowledge from data in high performance, and flexible means the system can deal with very large, heterogeneous distributed and complex data.

Contributions. This paper makes the following contributions:

- Provide a brief overview of DDM issues
- Describe a new DDM system that is easy to use, easy to extend and is flexible to facilitate seamless integration of distributed resources to deal with very large, heterogeneous distributed and complex data

The rest of this paper is organized as follows. DDM challenges will be discussed in the next section. Main DDM techniques will be presented and discussed in section 3. In section 4, the structure of ADMIRE will be shown. The paper ends with a short conclusion in section 5.

2 Challenges of DDM

In order to improve the performance of DDM technology and applications, and find practical solution of DDM system, we discuss main challenges of DDM in this section.

2.1 Distributed data

One of the main challenges of DDM is that distributed data is heterogeneous, complex and noisy. It's hard to deal with heterogeneous and complex data. Data in DDM can be divided into two categories: homogeneous and heterogeneous. In homogeneous DDM, the databases located at different sites have the same attributes and in the same format, while in heterogeneous DDM, the attributes at each site are different or in different format. Heterogeneous data is more complex than homogeneous data for DDM tasks.

Most studies on DDM assume that local databases are homogeneous. So many DDM algorithms only deal with homogeneous data. If the local databases are heterogeneous, we need to adopt different techniques to deal with them. Integrating local models of heterogeneous data is hard for many data mining tasks. Therefore, developing DDM algorithms that can handle heterogeneous data is becoming increasingly important.

Many real data are high dimensional, high dense, non static, unbalanced. Increasingly complex data sources, structures, and types (like natural language text, images, time series, continuous data streams, multi-relational and object data types etc.) are emerging. It requires the development of new methodologies, algorithms, tools, and services to mine such complex data. One solution for managing the complex data for DDM is to unify different data. For example, we can use XML to present complex data.

Sometimes, complexity of data rests with noise in the data. Real world data is dirty and noisy. In a large database, many of the attribute values will be inexact or incorrect, or there are some missing attributes and missing attribute values. Data noise may affect DDM results, so high quality data for DDM is needed. One solution is data preprocessing such as data cleaning, data transformation, data reduction.

2.2 Data privacy

We should consider data privacy and security when we use DDM technology to identify patterns and trends from large quantities of data. Privacy plays an important role in DDM. Data privacy in DDM assumes the data is distributed between two or more sites, and these sites cooperate to learn the global data mining results without revealing the data at their individual sites.

One of the earliest discussions about privacy in the context of data mining can be found in [1]. Survey [2] provides a detailed literature for privacy preserving data mining. The work in [3] proposes a paradigm for clustering distributed privacy sensitive data in an unsupervised or a semi-supervised scenario. Several

other distributed algorithms, e.g., the meta-learning approach [4], the Fourier spectrum-based decision tree integration approach [5], and the collective principal component analysis-based clustering algorithm [6], are also potentially suitable for privacy preserving mining from multi-party data. A distributed privacy-preserving algorithm for Bayesian network parameter learning is reported in [7, 8]. The work in [9] explores the possibility of using multiplicative random projection matrices for privacy preserving distributed data mining.

2.3 Distributed environment

Distributed environment is the base of DDM system. DDM needs effective infrastructures for distributed large-scale and high-performance computing and data processing. Various wired and wireless networks offer the distributed computing environment. Recently, grid is considered as more and more important distributed computing environment in DDM.

In centralized data mining, the main concern for the efficiency of a data mining algorithm is its I/O and/or CPU time. In a distributed environment, the communication cost should be considered, it may be a bottleneck in DDM [10, 11]. The cost of transferring large blocks of data may be prohibitive and result in very inefficient implementations in DDM. For a slow network, the communication cost will dominate the overall cost. The communication cost is determined by the infrastructures of the distributed environment, the network bandwidth and the number of messages that are sent across the network. In order to reduce the communication cost, many DDM methods are used to minimize the number of messages sent. Some methods also attempt to load-balance across sites to prevent performance from being dominated by the time and space usage of any individual site. We consider that one important method is to choose a suitable distributed infrastructures and computing service. The distributed computation infrastructure of grid is very suitable for DDM. Grid can provide an effective computational support for DDM applications.

A grid is a geographically distributed computation infrastructure composed of a set of heterogeneous machines that users can access via a single interface. A grid environment provides high performance computing facilities and transparent access to them in spite of their remote location, different administrative domains and hardware and software heterogeneous characteristics. Grid computing provides a novel distributed environment, computational model, and unprecedented opportunities for unlimited computing and storage resources. It's distinguished from conventional distributed computing by its focus on large-scale resource sharing, innovative applications, and, in some cases, high-performance orientation. Grids can be used as effective infrastructures for distributed high-performance computing and data processing [12].

DDM on grid, although is a fairly new research topic, has been very active in data mining community. There are some applications of grid for data mining such as [13, 14]. The main disadvantage of grid is that grid software and standards are still evolving. The development of DDM on grid isn't easy.

Peer-to-peer (or abbreviated P2P) architecture is a type of network in which each workstation has equivalent capabilities and responsibilities. This differs from client/server architectures. Generally, P2P networks are used for sharing files, but a P2P network can also mean Grid Computing. Techniques and applications of P2P for DDM can be found in [15].

The primary disadvantage of P2P is the tendency of computers at the edge of the network to fade in and out of availability. Also, accountability for the actions of network participants could be a difficult problem. Several high-profile implementations have shown that architecture, security, and systems management issues are difficult to control. For these reasons, system managers often prefer to operate P2P systems as separate isolated entities. But, doing so is often impossible for practical applications.

2.4 Distributed algorithm

Distributed algorithm is the core of DDM. We should consider above challenges when a distributed algorithm is designed. Efficient distributed algorithms and techniques need to be developed to deal with distributed data mining tasks and facilitate seamless integration of distributed resources for complex problem solving.

Most DDM algorithms are developed to mine distributed data in each site, and produce one local model per site. Subsequently all local models are aggregated to produce the global model. Many DDM algorithms for distributed association rule mining [16–18, 15, 19], distributed classification [20, 21] and distributed clustering [22–24, 3, 25] etc. have been presented in literatures.

Many existing DDM algorithms have their limitations, they can be efficient only for some specific cases. They suffer from huge, heterogeneous, high dimensional, high dense, noisy and complex data. So development of efficient distributed algorithms is always a main challenge for DDM.

3 DDM techniques

The increasing demand to scale up to massive data sets inherently distributed over a network with limited bandwidth and computational resources available motivated the development of the techniques of DDM. A number of approaches and techniques have been proposed in literatures. The books [26] and the survey [27] introduce the state-of-art techniques of DDM.

Some data mining techniques can be used to adapt DDM. Bayesian methods were developed in the framework of statistics for many years. Last ten years, they were applied in the problems of data mining. Bayesian methods in DDM are reported in [28, 29]. Decision tree is well-known in data mining. Decision tree technique has been used in DDM [30, 31]. Some statistical techniques such as bagging [32], boosting [33] and stacking [34] etc. could be extended to combine local models in a distributed environment. The techniques such as Multi-agent

Systems, ensemble learning, similarity-based [35, 36] and collective data mining [26, 10, 29] are presented in DDM literatures.

In this section, we mainly present the DDM techniques based on Multi-agent Systems and ensemble learning.

3.1 Agent-based

Multi-agent Systems (MAS) is the emerging subfield of artificial intelligence that aims to provide both principles for construction of complex systems involving multiple agents and mechanisms for coordination of independent agents' behaviors. MAS is fundamentally designed for collaborative problem solving in distributed environments. An agent-based data mining system is a natural choice for mining large sets of inherently distributed data. Many DDM systems such as PADMA [37], BODHI [38] and JAM [4], are based on multi-agent techniques.

The resource-constrained distributed environments of DDM and the need for collaborative approach to solve many of the problems in this domain make multi-agent systems-architecture an ideal candidate for application development. The power of multi-agent-systems can be further enhanced by integrating efficient data mining capabilities and DDM algorithms may offer a better choice for multi-agent systems since they are designed to deal with distributed systems.

Agents in MAS need to be pro-active and autonomous. Agents perceive their environment, dynamically reason out actions based on conditions, and interact with each other. In some applications the knowledge of the agents that guide reasoning and action depend on the existing domain theory. However, in many complex domains this knowledge is a result of the outcome of empirical data analysis in addition to pre-existing domain knowledge. Scalable analysis of data may require advanced data mining for detecting hidden patterns, constructing predictive models, and identifying outliers, among others. In a multi-agent system this knowledge is usually collective. This collective intelligence of a multi-agent system must be developed by distributed domain knowledge and analysis of distributed data observed by different agents. Such distributed data analysis may be a non-trivial problem when the underlying task is not completely decomposable and computing resources are constrained by several factors such as limited power supply, poor bandwidth connection, and privacy sensitive multi-party data, among others.

Survey [39] offers a perspective on DDM algorithms in the context of MAS. It discusses broadly the connection between DDM and MAS. [40] reviews prominent approaches in the literature and presents a novel scheme for agent-based distributed data clustering.

3.2 Ensemble learning

There are two main advantages of DDM using ensembles. The first advantage can be obviously seen when the local model is much smaller than the local data: sending only the model thus reduces the load on the network and the network bandwidth requirement. The second one is that sharing only the model, instead

of the data, gains reasonable security for some organizations since it overcomes issues of privacy.

Bagging [32], boosting [33] and stacking [34] have been applied for DDM. Meta-learning is one of the most important approach of ensemble learning, it is particularly suitable for distributed data mining applications [41].

Meta-learning is a general method that facilitates the combining of models computed independently by the various machine learning programs and supports the scaling of large data mining applications.

Meta-learning is an approach for classification. With meta-learning, an ensemble of classifiers is used to get a global classifier from large and inherently distributed databases. The main idea of meta-learning is to execute a number of concept learning processes on a number of data subsets in parallel, and then to combine their collective results through another phase of learning. Initially, each concept learning task, also called base learner, computes a concept or base classifier that models its underlying data subset or training set. Next, a separate concept learning task, called meta learner, combines these independently computed base classifiers into a higher level concept or classifier, called meta classifier, by learning from a meta-level training set. This meta-level training set is basically composed from the predictions of the individual base-classifiers when tested against a separate subset of the training data, also called validation set. From their predictions, the meta-learner detects the properties, the behavior and performance of the base-classifiers and computes a meta-classifier that models the global data set. Meta-learning is scalable as well as portable and extensible, and amenable to direct execution in network computing environments. Distributed meta-learning techniques have been developed [41, 4]. Each site develops a classifier independently, these are used in concert to produce the global classifier results.

4 ADMIRE systems

Facing the challenges of DDM and rethinking the DDM techniques, we propose a new DDM system on Grid: ADMIRE. Combining dynamic efficient DDM techniques and Grid-based service system, ADMIRE defines a new DDM infrastructure on real-world large applications. There are three layers in ADMIRE: application layer, DDM layer and Grid layer (see figure 1). This system aims at the DDM problems in application such as large real data, heterogeneous data, complex data etc. DDM is a complex process to extract and discover global knowledge from distributed data sets. In order to achieve DDM tasks, we combine DDM techniques and distributed environment to implement ADMIRE system that is easy to use, easy to extend and well flexible.

The main features of ADMIRE system include:

- Providing dynamic, scalable and flexible DDM algorithms for extracting knowledge efficiently. We use various specific algorithms or approaches for the same task, because many algorithms or approaches may exert efficient performance on specific data (particular in size, density, and type etc.) and

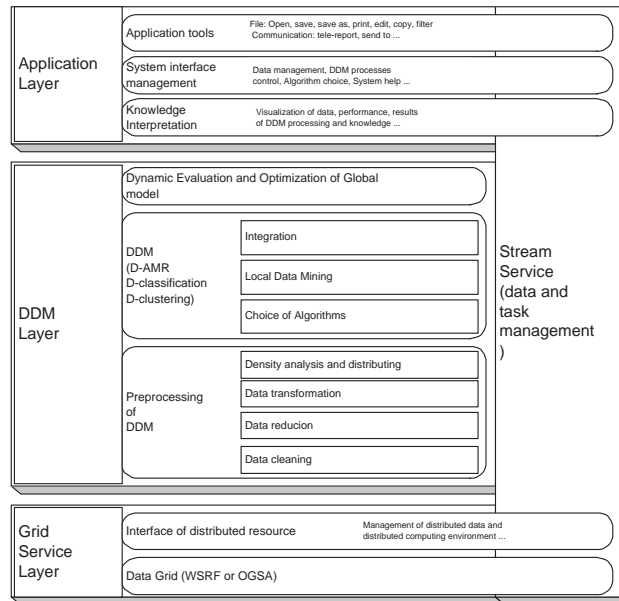


Fig. 1. The structure of ADMIRE.

distributed environment. The system should analyze the characteristics of the data, and then automatically choose a befitting algorithm. Different sites may perform different algorithms or approaches for the same task depending on different characteristics of each site.

- Focussing on the applications of real world data that are heterogeneous, high dimensional.
- Providing data preprocessing such as data analysis, distributing data at different sites, data cleaning, data transformation and data reduction.
- Based on Grid or P2P computing environment.
- Implemented in Java. Data is in XML format. With XML and Java, a DDM system is more easy to unify complex data and computing environment. Java ensures more security in DDM application.

4.1 DDM layer

The core of ADMIRE system is DDM layer. This layer outfits efficient DDM techniques and algorithms to deal with very large, heterogeneous distributed and complex data. All DDM techniques and algorithms in ADMIRE system are designed as plugins that are dynamic and flexible to be applied, swappable, and easy to be extended. The aim of the layer is to facilitate seamless integration of DDM algorithms, distributed data and distributed environment for complex problem solving. This layer contains 3 modules: DDM preprocessing, DDM, Dynamic Evaluation and Optimization of Global model.

DDM preprocessing is a very important process for ADMIRE system. Data preprocessing may facilitate DDM operations. It can ensure the quality of DDM results and improve the performance of DDM. According to the need, the system can preprocess data such as data cleaning, distributing data at different sites, data transformation, data reduction, data density analysis, etc.

DDM module includes Choice of Algorithms, Local Data Mining and Integration. According to the analysis of data density and data characteristic, we should choose one suitable efficient DDM algorithms and techniques, or a combination of some DDM algorithms and techniques for various DDM tasks to produce one local model per site. After local model has been produced at each site, all local models are integrated to produce the global model. Many DDM algorithms for distributed association rule mining, distributed classification and distributed clustering etc. will be applied in this module. Local data mining systems must adapt to work on local databases. In order to deal with heterogeneous, high dimensional and complex data, we use the following strategies:

- For the same task, different algorithms will be used to deal with different kind of data. Different algorithms and techniques may be used for different sites when we deal with the same task depending on different characteristics of each site. For example, there are lots of clustering algorithms, but most of them is only suitable for one type of data. Maybe many different types of data in the same or different sites for distributed data. Hence each site can use one suitable clustering algorithm or combination of several clustering algorithms.
- According to the application, we can unify some data mining tasks. For example, if we need both clustering and association rule mining for huge distributed transaction data, we can use some techniques to unify these two tasks for avoiding large amounts of repetitious computing.

The module of Dynamic Evaluation and Optimization of Global model is to identify and optimize dynamically the global patterns getting form DDM module. This module can supervise, evaluate and modulate the DDM processing and results.

Data privacy and security are considered in this layer when we use DDM technology to extract patterns and trends from large quantities of data.

4.2 Grid layer

Grid layer provides an efficient distributed computational platform and service for ADMIRE system.

4.3 Application layer

Application layer contains 3 modules: Application tools, System interface management and Knowledge Interpretation.

This layer is an user graphic interface to access the system to manage and perform various tasks. For example, we can achieve the operations: Open, Save,

Save as, Print, Edit, Copy, Filter, and so on in the module of Application tools. The module System interface management may help an user to manage distributed resource and operate DDM tasks such as preprocessing, distributed classification, distributed association rule mining, distributed clustering. The data, performance, results of DDM processing and knowledge may be represented in visualization.

There is a special module in ADMIRE system: Stream Service. This module provides the data management service and task execution service. Data stream and task stream run through the ADMIRE system, so we specify this special module for management of data and task.

5 Conclusion

Even if many techniques and systems of DDM have been proposed, huge and complex heterogeneous distributed data in the real world need us to develop more scalable and more efficient techniques for DDM, and practical applications of DDM require us to develop DDM system that is easy to use, easy to extend and very flexible. In order to develop new scalable and efficient DDM approach, this paper gives a brief overview of DDM issues and techniques. Furthermore, a new DDM system on grid: ADMIRE, is presented.

Ongoing research focus in particular on development of DDM technique and DDM system that can deal with huge, complex and heterogeneous distributed real world data.

Acknowledgements

This work is supported by the Marie Curie Fellowship Association, project ADMIRE.

References

1. Clifton, C., Marks, D.: Security and privacy implications of data mining. In of British Columbia Department of Computer Science, U., ed.: In Workshop on Data Mining and Knowledge Discovery, Montreal, Canada (1996) 15C19
2. Verykios, V.S., Bertino, E., Fovino, I.N., Provenza, L.P., Saygin, Y., Theodoridis, Y.: State-of-the-art in privacy preserving data mining. *SIGMOD Record* **33** (2004) 50–57
3. Merugu, S., Ghosh, J.: Privacy-preserving distributed clustering using generative models. In: The Third IEEE International Conference on Data Mining (ICDM'03), Melbourne, FL (2003)
4. Stolfo, S., et al.: JAM: Java Agents for Meta-Learning over Distributed Databases. In: Proceedings of Third International Conference on Knowledge Discovery and Data Mining, Menlo Park, CA, AAAI Press (1997) 74–81
5. Kargupta, H., Park, B.H.: A fourier spectrum-based approach to represent decision trees for mining data streams in mobile environments. *IEEE Transactions on Knowledge and Data Engineering* **16** (2004) 216C229

6. Kargupta, H., Huang, W., Sivakumar, K., Johnson, E.: Distributed Clustering Using Collective Principal Component Analysis. *Knowledge and Information Systems* **3** (2001) 422–448
7. Wright, R., Yang, Z.: Privacy-Preserving Bayesian Network Structure Computation on Distributed Heterogeneous Data. In: *Proceedings of The Tenth ACM SIGKDD Conference (KDD'04)*, Seattle, WA (2004)
8. Meng, D., Sivakumar, K., Kargupta, H.: Privacy Sensitive Bayesian Network Parameter Learning. In: *Proceedings of The Fourth IEEE International Conference on Data Mining (ICDM'04)*, Brighton, UK (2004)
9. Liu, K., Kargupta, H., Ryan, J.: Multiplicative noise, random projection, and privacy preserving data mining from distributed multi-party data. *IEEE Transactions on Knowledge and Data Engineering* **18** (2005) 92–106
10. Bhat, P.B., Raghavendra, C.S., Prasanna, V.K.: Efficient collective communication in distributed heterogeneous systems. *Journal of Parallel and Distributed Computing* **63** (2003) 251–263
11. Provost, F.: Distributed Data Mining: Scaling Up and Beyond. In Kargupta, H., Chan, P., eds.: *Advances in Distributed Data Mining*. MIT/AAAI Press (2000)
12. Chervenak, A., Foster, I., Kesselman, C., Salisbury, C., Tuecke, S.: *The Data Grid: Towards an Architecture For the Distributed Management and Analysis of Large Scientific Datasets* (1999)
13. Cannataro, M., Talia, D., Trunfio, P.: KNOWLEDGE GRID: High Performance Knowledge Discovery on the Grid. *GRID 2001* (2001) 38–50
14. Mastroianni, C., Talia, D., Trunfio, P.: Managing Heterogeneous Resources in Data Mining Applications on Grids Using XML-Based Metadata. In: *IPDPS*, Nice, France (2003)
15. Wolff, R., Schuster, A.: Association Rule Mining in Peer-to-Peer Systems . In: *Third IEEE International Conference on Data Mining*, Melbourne, FL (2003)
16. Cheung, D.W., Ng, V.T., Fu, A.W., Fu, Y.: Efficient Mining of Association Rules in Distributed Databases. *IEEE Transactions On Knowledge And Data Engineering* **8** (1996) 911–922
17. Vaidya, J., Clifton, C.: Privacy Preserving Association Rule Mining in Vertically Partitioned Data. In: *The Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Edmonton, Canada (2002)
18. Coenen, F., Leng, P., Shakil, A.: T-trees, Vertical Partitioning and Distributed Association Rule Mining. In: *The Third IEEE International Conference on Data Mining (ICDM'03)*, Melbourne, FL (2003)
19. Schuster, A., Wolff, R., Trock, D.: A High-Performance Distributed Algorithm for Mining Association Rules . In: *Third IEEE International Conference on Data Mining*, Florida , USA (2003)
20. Kotecha, J.H., Ramachandran, V., Sayeed, A.M.: Distributed Multitarget Classification in Wireless Sensor Networks. *IEEE Journal of Selected Areas in Communications* **23** (2005) 703–713
21. Basak, J., Kothari, R.: A Classification Paradigm for Distributed Vertically Partitioned Data. *Neural Computation* **16** (2004) 1525–1544
22. Forman, G., Zhang, B.: Distributed Data Clustering Can Be Efficient and Exact. *SIGKDD Explorations* **2** (2000) 34–38
23. Li, T., Zhu, S., Ogihara, M.: Algorithms for Clustering High Dimensional and Distributed Data. *Intelligent Data Analysis Journal* **7** (2003)
24. Klusch, M., Lodi, S., Moro, G.L.: Distributed Clustering Based on Sampling Local Density Estimates. In: *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI 2003)*, Mexico (2003) 485–490

25. Tsoumakas, G., Angelis, L., Vlahavas, I.: Clustering Classifiers for Knowledge Discovery from Physically Distributed Databases. *Data and Knowledge Engineering* **49** (2004) 223–242
26. Kargupta, H., Chan, P.: *Advances in Distributed and Parallel Knowledge Discovery*. AAAI/MIT Press (2000)
27. Zaki, M.: *Parallel and Distributed Association Mining: A Survey*. IEEE Concurrency (1999)
28. Sivakumar, K., Chen, R., Kargupta, H.: Learning Bayesian Network Structure from Distributed Data. In: *Proceedings of the 3rd SIAM International Data Mining Conference, San Francisco, CA* (2003) 284–288
29. Chen, R., Sivakumar, K., Kargupta, H.: Collective Mining of Bayesian Networks from Distributed Heterogeneous Data. *Knowledge and Information Systems* **6** (2004) 164–187
30. Kargupta, H., Park, B.: Mining Decision Trees from Data Streams in a Mobile Environment. In: *Proceedings of the IEEE International Conference on Data Mining, IEEE Press* (2001) 75–82
31. Giannella, C., Liu, K., Olsen, T., Kargupta, H.: Communication Efficient Construction of Decision Trees Over Heterogeneously Distributed Data. In: *Proceedings of The Fourth IEEE International Conference on Data Mining (ICDM'04), Brighton, UK* (2004)
32. Chawla, N.V., Moore, T.E., Hall, L.O., Bowyer, K.W., Kegelmeyer, W.P., Springer, C.: Distributed Learning With Bagging-like Performance. *Pattern Recognition Letters* **24** (2003) 455–471
33. Lazarevic, A., Obradovic, Z.: The Distributed Boosting Algorithm. In: *Knowledge Discovery and Data Mining*. (2001) 311–316
34. Tsoumakas, G., Vlahavas, I.: Effective Stacking of Distributed Classifiers. In: *Proceedings of the 15th European Conference on Artificial Intelligence*. (2002) 340–344
35. Li, T., Zhu, S., Ogihara, M.: A New Distributed Data Mining Model Based on Similarity. *ACM SAC Data Mining Track* (2003)
36. Parthasarathy, S., Ogihara, M.: Exploiting Dataset Similarity for Distributed Mining. In: *3rd Workshop on High Performance Data Mining. In conjunction with International Parallel and Distributed Processing Symposium 2000 (IPDPS'00), Cancun, Mexico* (2000)
37. Kargupta, H., Hamzaoglu, I., Stafford, B., Hanagandi, V., Buescher, K.: *Proceedings of the high performance computing conference*. In: *PADMA: Parallel Data Mining Agents For Scalable Text Classification*. (1997) 290–295
38. Kargupta, H., Hamzaoglu, I., Stafford, B.: Scalable, Distributed Data Mining Using An Agent Based Architecture. In Heckerman, D., Mannila, H., Pregibon, D., Uthurusamy, R., eds.: *Proceedings of Knowledge Discovery And Data Mining, Menlo Park, CA, AAAI Press* (1997) 211–214
39. Costa Da Silva, J., Giannella, C., Bhargava, R., Kargupta, H., Klusch, M.: Distributed data mining and agents. *International Journal of Engineering Applications of Artificial Intelligence* **18** (2005)
40. Klusch, M., Lodi, S., Moro, G.L.: Agent-Based Distributed Data Mining: The KDEC Scheme. In: *Intelligent Information Agents: The AgentLink Perspective*. LNAI 2586. Springer (2003) 104–122
41. Prodromidis, A., Chan, P.: Meta-learning in Distributed Data Mining Systems: Issues and Approaches. In Kargupta, H., Chan, P., eds.: *Advances of Distributed Data Mining*. MIT/AAAI Press (2000)