Distributed data mining: a survey

Li Zeng · Ling Li · Lian Duan · Kevin Lu ·
Zhongzhi Shi · Maoguang Wang · Wenjuan Wu ·
Ping Luo

Published online: 17 May 2012

Abstract Most data mining approaches assume that the data can be provided from a single source. If data was produced from many physically distributed locations like Wal-Mart, these methods require a data center which gathers data from distributed locations. Sometimes, transmitting large amounts of data to a data center is expensive and even impractical. Therefore, distributed and parallel data mining algorithms were developed to solve this problem. In this paper, we survey the-state-of-the-art algorithms and applications in distributed data mining and discuss the future research opportunities.

Keywords Data mining · Business intelligence · Business analytics · Decision support systems · Distributed systems · Literature review

1 Introduction

Traditional data mining algorithms assume that the data is centralized, memory-resident, and static [6, 9–11, 21, 29, 30, 32, 37, 40, 43, 45, 53, 56, 57]. However, this assumption is no longer valid with the development of Internet where data mining techniques meet two challenges. First, the amounts of data are generated too fast to be processed even for supercomputers. Second, the data was stored at multiple locations and it becomes increasingly expensive to centralize it in one place. Bandwidth limitation and privacy concerns are also the factors to hinder data centralization. To solve the above problems, Distributed Data Mining (DDM) has become a hot research area [34, 61, 62]. Distributed data mining has become popular as business intelligence market is one of fastest growing and most profitable areas in software industry. This paper is a brief survey of distributed data mining with emphasis on approaches and taxonomy in different environments.

DDM makes the assumption that either the computation or the data is distributed. It can be used in parallel supercomputers, P2P networks, and sensor networks. Under different situations, it has communication, privacy, and resource constraints such as computing power. In addition, distributed data mining is a process involving the application of specific algorithms; that is, it is hard to provide a unified framework for algorithms. Therefore, we categorize the existing work according to different computing and storage settings. Section 2 describes different distributed data mining techniques and applications according to the taxonomy in different environments. We discuss the future research opportunities and conclude the paper in Sect. 3.

2 Overview of distributed data mining

Distributed data mining can be used in parallel supercomputers, P2P networks, and sensor networks. However, different environments have different concerns. Distributed

L. Zeng · L. Duan · Z. Shi · M. Wang · P. Luo
Institute of Computing Technology,
Chinese Academy of Sciences, Beijing 100080, China

L. Li
Old Dominion University, Norfolk, VA 23529, USA

L. Duan (✉)
New Jersey Institute of Technology, Newark, NJ 07102, USA
e-mail: duanlian.cn@gmail.com

K. Lu
Brunel University, Uxbridge UB8 3PH, UK

W. Wu
School of Information, Remin University of China,
Beijing 100872, China

data mining techniques can be categorized into following types.

- Centered clusters versus peer-to-peer: The centered cluster has a coordinator. The coordinator splits the work to multiple computers. The centered cluster is easier to employ and coordinate. However, it involves the fair partition of work and may have a problem of single point of failure. It requires stable environments and usually employ in the supercomputing environment. Recently, peer-to-peer computing has very successful applications, such as peer download. It gathers users to join the service quickly and is considered as the most scalable system. Peer-to-peer algorithms do not depend on a central server, and each site gets the data and performs its own task. The nature of the peer-to-peer system is decentralization and each site has limited view to the entire system. This limit actually provides better security since sites do not need to observe irrelevant surroundings. In addition, light-weight algorithms can be transmitted across the network rather than the large amount of data. For example, site S1 has the algorithm A1 and site S2 needs to mine the data using A1. Then, transferring A1 from S1 to S2 is a better way than transfer all the data from S2 to S1. However, the security is an important issue in peer-to-peer systems since they exchange information with each other. In addition, some peers might only use others' resources while not providing service to others. Therefore, each peer must agree on terms and conditions of use before joining the system.
- Single model versus meta-learning: Single model partitions a given data mining algorithm into smaller parts distributed to each site. The algorithm can choose to move data, intermediate results, predictive models, and the final results of a data mining algorithm. Single model systems employ local learning models at each site and move the models to a central location. The coordination unit gathers the intermediate results from each site and generates the final results for the given algorithm. On the other hand, meta-learning [42], loosely defined as learning from learned knowledge, is another technique developed that deals with the problem of computing a "global" model from large and inherently distributed databases. The goal of meta-learning is to compute a number of independent model (classifiers) by applying learning programs in parallel without transferring or directly accessing the data sites. Such systems have more flexibility to select and combine different data mining models according to the relevance of different data sources.
- Homogeneous data versus heterogeneous data: In a distributed relational database system, different information might be saved in different sites. If each site has the complete relational tables, the homogenous data is saved for each site. If each site saves information for different tables, then we have the heterogeneous data issue. Most existing distributed data mining considers homogeneous data across different sites. In the case of heterogeneous distributed data, we only observe the incomplete knowledge about the complete data set. Different local models have the local view of the entire problem and distributed data mining has to generate a global model from those different local models. Therefore, mining heterogeneous data is challenging.

Despite of the above different categorization, we will introduce the distributed data mining algorithms and applications according to different system settings. First, some distributed data mining algorithms are designed according to the existing system needs and requirements instead of building the corresponding system setting to run a particular distributed data mining algorithm. Second, different system settings have different characteristics which require special consideration on different computing, communication, and storage sources when designing the corresponding distributed data mining algorithm.

2.1 Multi-core supercomputers

Under this setting, all the computing units share the same memory and the communication cost among computing units is negligible. Any data mining algorithm can be rewritten on the fine-grained parallelism level. That is, we allow the subtasks to communicate many times per second. They can also run coarse-grained parallelism algorithms where subtasks communicate occasionally or embarrassing parallelism algorithms where they rarely or never communicate. The modification of the original data mining algorithm is more related to traditional parallelism computing techniques [18] which can be applied to any type of algorithms. However, such type of devices are very expensive, and we usually will not consider this option unless the data mining algorithm is computationally very expensive and can only be rewritten as an efficient fine-grained parallel algorithm.

Another trend on this type of setting is the graphics processing unit (GPU) based parallel techniques. More and more supercomputers have the GPU component which is different from the traditional CPU [51]. Architecturally, a CPU is composed of a few cores with a relatively large cache memory to handle a few threads at a time, while a GPU is composed of hundreds of cores that can handle thousands of threads simultaneously. GPU is a programmable computational device with massive cores. It was originally used for 3D game rendering, but now its capabilities are being used to accelerate computational workloads including data mining. Different from multicore CPUs, the GPU threads are managed by the hardware and optimized for a huge batch of data taking the same operation simultaneously which is very fast for matrix operation. It is unnecessary and impossible to let the GPU run the whole data mining algorithm; however, for simple and repeat operations in the data mining algorithms like counting the occurrence of items from a bitmap [13] and search k-nearest neighbors [33], GPU can speed up the process significantly. Fang et al. [12] proposed a GPU-based architecture for data mining tasks. The architecture makes use of three components: [1] a CPU-based storage and buffer manager to handle the whole program flow and data transferring between CPU and GPU; [2] a GPU-CPU co-processing mining module; and [3] a GPU-based visualization module. They successfully implemented k-means clustering [63] and the Apriori frequent itemset mining algorithms [60] in this framework.

2.2 Peer-to-peer based

Communication bandwidth is often a scarce resource for data mining in decentralized circumstances. Approaches collecting all local data at one given node (or central node) will result in communication bottlenecks or messages implosion somewhere. Remarkably, peer-to-peer (P2P) networks provide a natural and well-suited platform for such distributed data mining, as well as information dissemination, file sharing, and e-business. As mentioned, DDM has introduced the distribution versions of many traditional data mining algorithms, such as association rule mining, clustering, and classification. However, most DDM efforts assume stable networks and data, and they cannot be applied directly to dynamic topology for the possible occurrence of node failure, link failure, and immediate join-leave behavior. In distributed data mining over P2P networks, most recent researches focus on developing local algorithms for primitive operations such as random sampling and aggregate functions.

The most representative example of peer-to-peer distributed data mining is distributed association rule mining [60] and distributed decision tree [28]. Kargupta et al. [22, 23] first proposed collective data mining which works on vertically partitioned data and combines immediate results from local data sources. Several different methods have been employed to combine predictive models built at different sites. Meta-learning combines several models by building a separate meta-model with inputs that are the outputs of the various models and the output is the desired outcome [20, 48]. Multiple models, which are often called ensembles of models, have been used for quite a while in centralized data mining. Methods for combining models in an ensemble include Bayesian model averaging for regression models [44], partition learning [19]. The related systems include JAM system developed by Stolfo et al. [48], Kensington system [20], and Papyrus developed by Bailey et al. [19]. JAM employs meta-learning, while Kensington uses knowledge probing. JAM is a distributed, scalable and portable agent-based data mining system that provides a set of meta-learning agents for combining multiple models that were learned at different sites. As long as a machine learning program is defined and encapsulated as an object conforming to the interface requirements, it can be imported and used directly in JAM. This plug and play characteristic makes JAM truly powerful and extensible data mining facility possible, although using a centralized approach to maintain the global configuration may obviously be a sequential bottleneck. JAM is so far the most representative system to mine distributed databases by means of meta-learning and intelligent agents. Papyrus is designed to support different data, tasks, and model strategies. In contrast to JAM, Papyrus requires not only to move models from node to node, but also to move data from node to node. Also, Papyrus is a specialized system which is designed for clusters, meta-clusters, and super-clusters, while most systems are designed only for mining data distributed over the Internet.

Kowalczyk et al. [26] used a model to estimate the mean value of distributed data, and provided the theoretical and experimental evidence for its feasibility to distributed data mining over P2P networks. The approach proposed by Kempe [24] relied on uniform gossip-based randomized algorithms for computing aggregate information.

Albashiri et al. [2] proposed an extendible multi-agent data miner system (EMADS) comprising a set of agents in a set of containers. The system has data agents, user agents, task agents, mining agents, and housekeeping agents. The main container maintains the housekeeping agents that provide facilities to execute the function of EMADS, a management system agent which controls the life cycles of other agents, and a directory facilitator agent which provides an agent lookup service.

Mehyar et al. [38] proposed a class of asynchronous distributed algorithms, which is a Laplacian-based approach to average the local inputs of nodes on a P2P network. Remarkably, their algorithm did not rely on synchronization, coordinated parameter values, and the apriori knowledge of global topology. Datta et al. [7, 8] presents an overview of efforts to develop DDM application and algorithms in P2P networks, and introduce P2P data mining algorithms that work in a decentralized manner. Wolff et al. [54, 55] introduced a local algorithm, and proposed an algorithm for monitoring K-means clustering in P2P networks. Similar work by Babcock et al. [3]

assumed a centralized coordinator site and a hierarchical topology for global conflict resolution, and proved that distributed communication is only necessary occasionally. Zhou et al. [64] proposed a distributed text mining system with the storage layer, the basic mining layer, and the analysis service layer. Messages are passed through layers and the basic mining layer makes use of asynchronous communication between multiple instances and map-reduce model to allocate computing resource closer to the place where data is stored.

2.3 Internet and gird computing based

Internet and grid computing seek to change the way we tackle complex problems [5, 16, 50]. Internet provides not only huge volume of distributed data, but also plenty distributed computing resources to construct a supercomputer than ever before [17]. Complex and data-intensive computing such as bioinformatics data mining, climate prediction models or airplane computer-aided design systems used to run on supercomputers only. Recently, however, rapid improvements in Internet, distributed algorithms, and the increase in speed and memory of the PC machine, lead to consider more decentralized approaches to computing. There are over billions of PCs around the world, and most are idle much of the time. Internet computing exploits these idle workstations and PCs to create powerful distributed computing systems with global reach. It has been possible to undertake many significant projects required supercomputer capabilities before on normal and inexpensive equipments. Some examples of popular internet computing projects include:

- Models@Home for Distributed Computing in Bioinformatics [27]
- SETI@home for Massively Distributed Computing [52]
- FightAIDSatHome for Fundamental Research in discovering New Drugs [14]
- ClimatePrediction for Simulating Earth's Climate [46, 48]

Projects such as Models@Home for distributed computing in bioinformatics using a screensaver based approach, SETI@Home the world's largest distributed computing project to detect intelligent life outside Earth, have demonstrated the principles and techniques of distributed computing which provides a mechanism to tap into the idle resource of millions of distributed PCs. Other project such as FightAIDSatHome is the first biomedical distributed computing project ever launched. It is run by the Olson Laboratory at the Scripps Research Institute in California. FightAIDSatHome uses scattered computer's idle resources to assist fundamental research in discovering new drugs, building on growing knowledge of the structural biology of AIDS. These projects require participating machines to download client data and models that execute on local machines, and then upload the result to servers. The ClimatePrediction project developed by Stainforth et al. [46] takes the distributed computing paradigm a step further by inviting participants to download a full-scale climate model and run it locally to simulate 100 years of the Earth's climate. Each participant independently carries out a subset of result which is then combined with other participants' results to ensemble a global climate prediction result. Obviously, low communication overhead was generated during the running as each participant has the complete model.

A grid is a geographically distributed computation infrastructure composed of a set of heterogeneous machines that users can access via a single interface. Grids provide common resource-access technology and operational services across widely distributed boundary. Grid computing therefore is described as seamless, pervasive access to resources and services [58, 59]. It plays a significant role in providing an effective computational support for application of distributed data mining. Grid computing has been proposed as a novel computational model, distinguished from conventional distributed computing by its focus on large-scale resource sharing, innovative applications, and, in some cases, high-performance orientation. Today, grids can be used as effective infrastructures for distributed high-performance computing and data processing [4, 15]. Cannataro et al. [4] discussed how to design and implement data mining applications by using the KNOWLEDGE GRID tools starting from searching grid resources, composing software and data components, and executing the data mining process on a grid. Talia [49] proposed a two-phase scheduling framework including external scheduling and internal scheduling in a two-level grid environment, upon which a system named DMGCE (Data Mining Grid Computing Environment) has been developed and implemented. Stankovski et al. [47] develop a DataMiningGrid architecture with three layers. The bottom layer manages grid software, data, and hardware resources. The middle layer provides middleware function, such as virtual organization management, resource management, job scheduling, security, to the entire system. The top layer summarizes the results and provides the final output to users.

Service-oriented architecture (SOA) provides seamless integration of self-contained computational services that can communicate and coordinate with each other to perform goal-directed computation [36, 39]. The concept has blossomed in the past few years due to the development of Web service related standards and technologies, including WSDL, Universal Description, Discovery, and Integration (UDDI), and SOAP. WEKA4WS [49], an open-source framework derived from the Weka toolkit for supporting distributed data mining on Grid environments, exposes all the data mining algorithms provided by as WSRF-compliant services (WSRF: Web Services Resource Framework), which enable important benefits such as dynamic service discovery and composition, standard support for authorization and cryptography, etc. Grid-enable Weka [25] is another proposed system found on Weka, which is a widely used toolkit for machine learning and data mining written in Java. In Grid-enable Weka, the execution of these tasks can be distributed across computers in an ad hoc Grid environment. Tasks that can be executed using Grid-enable Weka include building a classifier on a remote machine, testing a classifier on a dataset, and cross validation. Grid-enable Weka modifies the Weka toolkit to enable the use of multiple computational resources when performing data analysis. WekaG [41] is another system derived from Weka. It is based on client/server architecture. The server side provides a set of services that implement the functionalities and the process of the different data mining algorithms. The client side is responsible for communicating with server side and offering user interface. FAEHIM (Federated Analysis Environment for Heterogeneous Intelligent Mining) is a web services-based toolkit for supporting distributed data mining. This toolkit consists of a set of data mining services, a set of tools to interact with these services, and a workflow system used to assemble these services and tools.

3 Conclusions

Both distributed data mining and parallel data mining can speed up the data mining process, however, they are different in several ways. Distributed data mining often applies the same or different mining algorithms to tackle local data and communicate among multiple process units, and then combine the local pattern discovery by local data mining algorithms from local databases into a global knowledge solution. In this case, the discovered knowledge is often different from the knowledge discovered by applying the data mining algorithms to the entire dataset. The accuracy or efficiency of distributed data mining is somewhat difficult to predict, because it depends on data partitioning, task scheduling, and global synthesizing [1]. In contrast to distributed data mining, a parallel data mining algorithm discovers the same knowledge as that found by its sequential algorithm due to applying the global parallel algorithm on the entire data set. Its accuracy may be more guaranteed than the distributed data mining.

Distributed data mining enables geographically distributed science and engineering teams to collaborate in new ways. Despite recent advances in peer-to-peer and service-oriented technologies, grid computing, high-performance computing, issues on distributed computing such as limitation on data transmission bandwidth, skew distribution, very-large data size, privacy preservation [35] still need serious and immediate attention. For example, the execution environments of geographically distributed applications are far less deterministic than those of locally centralized. In addition, distributed applications become highly complex when large-scale and skewed data distributes in unpredictable locations in which identifying and correcting performance bottlenecks exposed may not be the same, because the network bandwidths and computing resources often vary from one place to another.

Despite advancements in the field of distributed data mining, both in academia and industry, significant challenges still remain. They need to be dealt with in order to fully realize the potential of distributed data mining. Distributed data mining will continue to embrace cutting-edge technology and techniques and will open up new applications that will impact business intelligence, business analytics, and industrial information integration engineering (IIIE) [31, 59].

Acknowledgments This work is partially supported by the Chinese Academy of Sciences under Grant No. 20040402, Changjiang Scholar Program of the Ministry of Education of China, National Natural Science Foundation of China under Grant No. 71132008, US National Science Foundation under Grant No. 1044845.

References

1. AbdelSalam H, Maly K, Mukkamala R, Zubair M, Kaminsky D (2010) Scheduling-capable autonomic manager for policy-based IT change management system. Enterp Inf Syst 4(4):423–444
2. Albashiri K, Coenen F, Leng P (2009) EMADS: an extendible multi-agent data miner. Knowl Based Syst 22(7):523–528
3. Babcock B, Olston C (2003) Distributed top-K monitoring. In: Proceedings of the 2003 ACM SIGMOD international conference on management of data (SIGMOD ‘03). ACM, New York, NY, USA, pp 28–39
4. Cannataro M, Congiusta A, Pugliese A, Talia D, Trunfio P (2004) Distributed data mining on grids: services, tools, and applications. IEEE Trans Syst Man Cybern B Cybern 34(6):2451–2465
5. Cao X, Yang F (2011) Measuring the performance of internet companies using a two-stage data envelopment analysis model. Enterp Inf Syst 5(2):207–217
6. Chiang D, Lin C, Chen M (2011) The adaptive approach for storage assignment by mining data of warehouse management system for distribution centres. Enterp Inf Syst 5(2):219–234
7. Datta S, Bhaduri K, Giannella C, Wolff R, Kargupta H (2006) Distributed data mining in peer-to-peer networks. IEEE Internet Comput 10(4):18–26
8. Datta S, Giannella C, Kargupta H (2006) K-means clustering over large, dynamic networks. In: Proceedings of 2006 SIAM conference data mining (SDM 06). SIAM Press, 2006, pp 153–164
9. Duan L, Xu L, Guo F, Lee J, Yan B (2007) A local-density based spatial clustering algorithm with noise. Inf Syst 32:978–986
10. Duan L, Xu L, Liu Y, Lee J (2009) Cluster-based outlier detection. Ann Oper Res 168:151–168

11. Duan L, Street W, Xu E (2011) Heathcare information systems: data mining methods in the creation of a clinical recommender system. Enterp Inf Syst 5(2):169–181
12. Fang W, Lau K, Lu M, Xiao X, Lam C, Yang Y, He B, Luo Q, Sander P, Yang K (2008) Parallel data mining on graphics processors. Technical Report, HKUST-CS08-07
13. Fang W, Lu M, Xiao X, He B, Luo Q (2009) Frequent itemset mining on graphics processors. In: Proceedings of the fifth international workshop on data management on new hardware (DaMoN '09). ACM, New York, USA, pp 34–42
14. Forli S (2011) Fight AIDS at Home Project, http://fightaidsathome.scripps.edu/, 2011
15. Foster I, Kesselman C, Tuecke S (2001) The anatomy of the grid: enabling scalable virtual organizations. Int J High Perform Comput Appl 15(3):200–222
16. Fu C, Zhang G, Yang J, Liu X (2011) Study on the contract characteristics of Internet architecture. Enterp Inf Syst 5(4): 495–513
17. Gong Z, Muyeba M, Guo J (2010) Business information query expansion through semantic network. Enterp Inf Syst 4(1):1–22
18. Kumar V, Grama A, Gupta A, Karpis G (2003) Introduction to parallel computing: design and analysis of parallel algorithms. Addison Wesley, Reading, MA
19. Grossman R, Bodek H, Northcutt D, Poor V (1996) Data mining and tree-based optimization. In: The proceedings of the second international conference on knowledge discovery and data mining (KDD-96). AAAI Press, MenloPark, California, pp 323–326
20. Guo Y, Ruger S, Sutiwaraphun J, Forbes-millott J (1997) Meta-learning for parallel data mining. In: Proceedings of the seventh parallel computing workshop, pp 1–2
21. Ingvaldsen J, Gulla J (2012) Industrial application of semantic process mining. Enterp Inf Syst 6(2):139–163
22. Kargupta H, Sanseverino E, Park B, Silvestre L, Hershberger D (1999) Scalable data mining from vertically partitioned feature space using collective mining and gene expression based genetic algorithms. KDD-98 workshop on distributed data mining
23. Kargupta H, Hoon B, Hershberger D, Johnson E (1999) Collective data mining: a new perspective towards distributed data mining. In: Kargupta H, Chan P (eds) Advances in distributed and parallel knowledge discovery. MIT/AAAI Press, Cambridge, MA, pp 133–184
24. Kempe D, Dobra A, Gehrke J (2003) Gossip-based computation of aggregate information. In: Proceedings of the 44th annual ieee symposium on foundations of computer science (FOCS '03). IEEE Computer Society, Washington, DC, USA, pp 1–10
25. Khoussainov R, Zuo X, Kushmerick N (2004) Grid-enabled weka: a toolkit for machine learning on the grid. ERCIM News No. 59, Oct 2004
26. Kowalczyk W, Jelasity M, Eiben A (2003) Towards data mining in large and fully distributed peer-to-peer overlay networks. In: Proceedings of 15th Belgian-Dutch conference on artificial intelligence (BNAIC 03). University of Nijmegen Press, pp 203–210
27. Krieger E, Vriend G (2002) Models@Home: distributed computing in bioinformatics using a screensaver based approach. Bioinformatics 18(2):315–318
28. Kubota K, Nakase A, Sakai H, Oyanagi S (2000) Parallelization of decision tree algorithm and its performance evaluation. In: Proceedings of the the fourth international conference on high-performance computing in the Asia-Pacific region, vol 2. IEEE, pp 574–579
29. Li H, Xu L (2001) Feature space theory—a mathematical foundation for data mining. Knowl Based Syst 14:253–257
30. Li H, Xu L, Wang J, Mo Z (2003) Feature space theory in data mining: transformations between extensions and intensions in knowledge representation. Expert Syst 20(2):60–71
31. Li L (2011) Introduction: advances in e-business engineering. Inf Technol Manage 12(2):49–50
32. Li L, Warfield J, Guo S, Guo W, Qi J (2007) Advances in intelligent information processing. Inf Syst 32(7):941–943
33. Liang S, Liu Y, Wang C, Jian L (2009) A CUDA-based parallel implementation of K-nearest neighbor algorithm. International conference on cyber-enabled distributed computing and knowledge discovery (CyberC'09), Oct 2009, Zhangjiajie, China, pp 291–296
34. Liu B, Cao S, He W (2011) Distributed data mining for e-business. Inf Technol Manage 12(2):67–79
35. Liu K, Kargupta H, Ryan J (2006) Random projection-based multiplicative data perturbation for privacy preserving distributed data mining. IEEE Trans Knowl Data Eng 18(1):92–106
36. Liu R, Deters R, Zhang W (2010) Architectural design for resilience. Enterp Inf Syst 4(2):137–152
37. Luo J, Xu L, Jamont J, Zeng L, Shi Z (2007) A flood decision support system on agent grid: method and implementation. Enterp Inf Syst 1(1):49–68
38. Mehyar M, Spanos D, Pongsajapan J, Low S, Murray R (2005) Distributed averaging on peer-to-peer networks. In: Proceedings of IEEE conference on decision and control. IEEE CS Press, 2005
39. Mietzner R, Leymann F, Unger T (2011) Horizontal and vertical combination of multi-tenancy patterns in service-oriented applications. Enterp Inf Syst 5(1):59–77
40. Perez-Castillo R, Weber B, Pinggera J, Zugal S, Guzman I, Piattini M (2011) Generating event logs from non-process-aware systems enabling business process mining. Enterp Inf Syst 5(3):301–335
41. Perez M, Sanchez A, Herrero P, Robles V, Pena J (2005) Adapting the weka data mining toolkit to a grid based environment. Lect Notes Comput Sci 3528:819–820
42. Prodromidis A, Chan P, Stolfo S (2000) Meta-learning in distributed data mining systems: issues and approaches. In: Advances in distributed and parallel knowledge discovery, vol 114. AAAI Press, p 38
43. Qian Y, Jin B, Fang W (2011) Heuristic algorithms for effective broker deployment. Inf Technol Manage 12(2):55–66
44. Raftery A, Madigan D, Hoeting J (1997) Bayesian model averaging for linear regression models. J Am Stat Assoc 92(437):179–191
45. Shi Z, Huang Y, He Q, Xu L, Liu S, Qin L, Jia Z, Li J, Huang H, Zhao L (2007) MSMiner-a developing platform for OLAP. Decis Support Syst 42(4):2016–2028
46. Stainforth D, Kettleborough J, Allen M, Collins M, Heaps A, Murphy J (2002) Distributed computing for public-interest climate modeling research. Comput Sci Eng 4(3):82–89
47. Stankovski V, Swain M, Kravtsov V, Niessen T, Wegener D, Kindermann J, Dubitzky W (2008) Grid-enabling data mining applications with datamininggrid: an architectural perspective. Future Gener Comput Syst 24(4):259–279
48. Stolfo S, Tselepis A, Lee W, Fan D, Chan P (1997) JAM: java agents for meta-learning over distributed databases. In: Proceedings of the third international conference on knowledge discovery and data mining (KDD-97). AAAI Press, Menlo Park, California, 1997
49. Talia D, Trunfio P, Verta O (2005) Weka4WS: a WSRF-enabled weka toolkit for distributed data mining on grids. In: Proceedings of the 9th european conference on principles and practice of knowledge discovery in databases. Porto, Portugal, pp 309–320
50. Tan W, Xu Y, Xu W, Xu L, Zhao X, Wang L, Fu L (2010) A methodology toward manufacturing grid-based virtual enterprise operation platform. Enterp Inf Syst 4(3):283–309
51. Top500.org (2011) Top 500 supercomputers. http://www.top500.org/list/2011/11/100
52. Werthimer D, Cobb J, Lebofsky M, Anderson D, Korpela E (2001) SETI@home: massively distributed computing for SETI. Comput Sci Eng 3(1):78–83

53. Wetzstein B, Leitner P, Rosenberg F, Dustdar S, Leymann F (2011) Identifying influential factors of business process performance using dependency analysis. Enterp Inf Syst 5(1):79–98
54. Wolff R, Schuster A (2004) Association rule mining in peer-to-peer systems. IEEE Trans Syst Man Cybern B Cybern 34(6): 2426–2438
55. Wolff R, Bhaduri K, Kargupta H (2006) Local L2-thresholding based data mining in peerto-peer systems. In: Proceedings of the 2006 SIAM conference data mining (SDM06). SIAM Press, pp 430–441
56. Xu L (2006) Advances in intelligent information processing. Expert Syst 23(5):249–250
57. Xu L, Liang N, Gao Q (2008) An integrated approach for agricultural ecosystem management. IEEE Trans SMC Part C 38(4): 590–599
58. Xu L (2011) Information architecture for supply chain quality management. Int J Prod Res 49(1):183–198
59. Xu L (2011) Enterprise systems: state-of-the-art and future trends. IEEE Trans Industr Inf 7(4):630–640
60. Zaki M (1999) Parallel and distributed association mining: a survey. IEEE Concurr 7(4):14–25
61. Zeng L, Lu K, Xu L, Shi Z, Luo P (2006) Distributed data mining: approaches and applications. Working paper, Institute of Computing Technology, Chinese Academy of Sciences
62. Zeng L, Xu L, Shi Z, Wang M, Wu W (2007) Distributed computing environment: approaches and applications. In: Proceedings of IEEE international conference on SMC 2007, Montreal, pp 3240–3244
63. Zhao W, Ma H, He Q (2009) Parallel K-means clustering based on mapreduce. In: Proceedings of the 1st international conference on cloud computing (CloudCom ‘09). Springer, Berlin, Heidelberg, pp 674–679
64. Zhou B, Jia Y, Liu C, Zhang X (2010) A distributed text mining system for online web textual data analysis. In: Proceedings of 2010 international conference on cyber-enabled distributed computing and knowledge discovery (CyberC), Oct 2010, pp 1–4