Improving Distributed Data Mining Techniques by Means of a Grid Infrastructure

Alberto Sánchez, José M. Peña, María S. Pérez, Víctor Robles, and Pilar Herrero

Facultad de Informática, Universidad Politécnica de Madrid, Madrid, Spain

Abstract. Nowadays, the process of data mining is one of the most important topics in scientific and business problems. There is a huge amount of data that can help to solve many of these problems. However, data is geographically distributed in various locations and belongs to several organizations. Furthermore, it is stored in different kind of systems and it is represented in many formats. In this paper, different techniques have been studied to make easier the data mining process in a distributed environment. Our approach proposes the use of grid to improve the data mining process due to the features of this kind of systems. In addition, we show a flexible architecture that allows data mining applications to be dynamically configured according to their needs. This architecture is made up of generic, data grid and specific data mining grid services.

Keywords: Data Mining, Grid Computing, Data Grid, Distributed Data Mining, Data Mining Grid.

1 Introduction

Data mining is characterized to be a complex process. There are two main characteristics that highlight this complexity. First, there are many non-trivial tasks involved in a standard data mining process. These tasks involve different activities like data preprocessing, rule induction, model validation and result presentation. A second determinant factor of data mining problems is the volume of the datasets they deal with.

Modern data mining systems are state-of-the-art applications that use advanced distributed technologies like CORBA, DCOM or Java-oriented platforms (EJB, Jini and RMI) to distribute data mining operations on a cluster of workstations or even all over the Internet. Distribution is a very important ally in the resolution of data mining problems. There are two main reasons to distribute data mining: (i) On the one hand, the efficient use of multiple processors to speed up the execution of heavy data mining tasks and (ii) On the other, there is originally distributed data that cannot be integrated into a single database due to technical or privacy restrictions.

The requirements of high performance data mining have been studied by some researchers. Maniatty, Zaki and others [26,38] collected the most important technological factors both hardware and software for data mining. Hardware support for redundant disks (RAID) and processor configurations (SMP computers and NUMA architectures) are mentioned. Within software contributions, parallel/distributed databases, parallel I/O and file systems are identified as appropriate data storages. Additional factors such as communication technologies like MPI (Message Passing Interface), CORBA, RMI or

R. Meersman et al. (Eds.): OTM Workshops 2004, LNCS 3292, pp. 111–122, 2004.

RPC are also important features. Data Space Transfer Protocol (DSTP) [2] is an example of specific communication protocol for distributed data mining.
The studies of different algorithms approaches like Provost [29] analyzed two main parallel and distributed schemas (fine-grain and coarse-grain) and their application to distributed data mining problems. Krishnaswamy defined cost models for distributed data mining [25]. Joshi et al. [17] provided an overview of different parallel algorithms for both association and classification rules. Another overviews can be found on: Kamath and Musick [18], Kargupta and Chan [21,20], Provost and Kolluri [30] and Zaki [35]. These survey contributions are worthwhile to understand the different solutions patterns adopted when a distributed data mining algorithm is designed and also to evaluate the results obtained.

Nevertheless, all these needs are not met by traditional and homogeneous distributed systems. Grid computing has emerged as a new technology, whose main challenge is the complete integration of heterogeneous computing systems and data resources with the aim of providing a global computing space [8]. We consider that grid provides a new framework in which data mining applications can be sucessfully deployed.

The paper is organized as follows. First, we present a overview of distributed data mining. Then, we analyze the requirements of these systems. Section 4 describes the current state of grid technology and its relation with data management. In Section 5, a new flexible and vertical architecture is proposed for data mining grid. Finally, we conclude with some remarks and the ongoing and future work.

2 Distributed Data Mining

Data mining evolution has outlined the development of new contributions in any of the following two lines: (i) New algorithms, theoretical models or data mining techniques and (ii) Technological and design research for new data mining systems and architectures. The same can be asserted for distributed data mining topic. New distributed algorithms, derived from the original centralized versions, have been developed. Some examples are distributed or parallel algorithms for association rules [33], classification rules [34], sequence patterns [36] or clustering algorithms [19]. The second research line is also very important because it deals with the efficient use of computational resources as well as with technical issues related to communication, synchronization, scheduling and so on. The behavior of data mining processing has specific usage patterns in terms of these technical aspects. These patterns have to be studied to design efficient distributed data mining systems.
There are many important advances of commercial products for data mining like IBM's Intelligent Miner [15], Clementine [32] from ISL/SPSS or SGI's MineSet [24] and others[1]. But there are only few commercial data mining systems that provide distributed data analysis at the present moment. A distributed version of Clementine [16] is the most representative example.

Distributed Data Mining Systems (DDM) is an innovative topic and currently only experimental system implementations are under development.

[1] An extended list of free and commercial data mining system can be found at `http://www.kdnuggets.com/software`

JAM (Java Agent for Meta-learning) [28] is an architecture developed at University of Columbia. JAM has been developed to gather information from sparse data sources and induce a global classification model. JAM technology is based on the meta-learning technique. Meta-learning makes it possible to build partial models. These partial models are combined in a global model using the meta-learning formal support. JAM system induces local models from different locations called, datasites. These models are exchanged among the datasites in order to combine them in a common model. The basic elements of JAM architecture are learning agents and meta-classifiers that are placed at each of the datasites to perform local tasks and to communicate the models between nodes. JAM system is a very interesting approach for distributed classification but this approach is not easy to translate into other data mining tasks and queries.

PADMA system [22] is being developed in Los Alamos National Laboratory. PADMA is a document analysis tool working on a distributed environment. The basic elements of this architecture are cooperative agents that analyze the text, interact with other agents and show the results to the user. PADMA system works without any relational database underneath. Instead, there are PADMA agents that perform several relational operations (for example join or selection) with the information extracted from the documents. PADMA system has been developed by using C++ and MPI interface. Documents are stored on an PPFS [14] file system.
Database integration is the key feature of PADMA system. The internal implementation of basic relational operations works tightly-coupled with the data analysis functions.

Papyrus [11] is a distributed data mining system developed by the University of Chicago, Illinois and the National Data Mining Center. Papyrus is able to mine distributed data sources on a WAN network scenario. Papyrus system uses meta-clusters to generate local models which are exchanged to generate a global model. The idea is founded on a theory similar to JAM system. Although JAM and Papyrus have common features, the second is a more advanced architecture in two senses: (i) It uses a formalized model representation language (PMML) to exchange local models, (ii) Data is managed by an efficient multi-layer storage system called Osiris which is an evolution from a persistent object library named PTool. Papyrus has been programmed over an Agent environment called Best which has been implemented in Agent-Tcl, an extended version of the script language Tcl. Additional features, like VRML presentation of results are also supported. Papyrus is a high performance system for distributed data mining, but it is also restricted (like JAM) to a subset of all the possible data mining queries. Its main contribution is the inclusion of a specialized data access system optimized for data mining tasks.

Kensington [23] was originally developed as a research project at London Imperial College and it is currently supported as a commercial tool. Kensington architecture is based on a distributed component environment located at different nodes on a generic network, like the Internet. Kensington provides three kind of components: (i) User oriented components, (ii) Application servers and (iii) Third level servers. The first group interacts with the user getting commands from him and showing the results. The application servers handle persistent objects, control sessions, schedule tasks and, in general, they manage the overall data analysis processes. The third level servers perform massive computation tasks, by retrieving data from the databases and processing them in high performance computation clusters (HPCs).

Kensington system has been developed using open technologies like EJB (Enterprise Java Beans) and CORBA. For HPCs tasks and algorithm implementations C code and MPI (Message Passing Interface) have been used. Nevertheless, DDM Systems require a new scenario in order to fulfill their requirements. Next section talks about these needs.

3 What Actually Does DDM Need?

This contribution comes as a result of the analysis, mentioned above, of technological features highlighted by Zaki and Maniatty as well as the study of many different distributed algorithms patterns. Distributed data mining, although is not completely different from the traditional distributed processing problems, it has special characteristics to bear in mind. The support of many algorithm patterns is the key feature to ensure the extensibility of new open DDM systems. Every existing and new algorithm has a specific combination of the following features:

- Data partitioning schema: Vertical vs Horizontal.
- Communication primitive: One-to-one, one-to-many or different variations.
- Synchronization and control aspects: From centralized to distributed control.
- Granularity and load balancing criteria.
- Sub–algorithm scope: local vs global.
- Sub–algorithm distribution: independent vs centralized.

Many other characteristics may also be included in the list. These are not only features of association, classification and clustering algorithms. Pre-processing tasks like data cleaning, attribute discretization, concept generalization and others should also be performed in parallel in the same way knowledge extraction algorithms do.

DDM systems must keep the existing features of the current centralized versions. The most important characteristic is the support for the complete KDD process: preprocessing, data mining and post-processing. The additional features either inherited from the centralized version or distributed specific are the following:

- For data mining operations, general purpose DBMS are less efficient than flat files [27], but they don't provide any data management features (e.g. relational operations). Specialized database managers are required (specially on a distributed environment). Relational and mining operations must be integrated and defined at the same level.
- State-of-the-art trends and high-performance technologies are very useful tools and become necessary when interoperable and efficient systems are designed. Open architectures like CORBA are oriented towards system interoperability with different scenarios, most of the private LAN networks, but scalability is an important trend to be achieved. Grid Computing provides these extreme parallelization features.
- Factors like the inclusion of algorithms, changes on system topology, different user privileges or system load restrictions have a deep impact in the overall performance of the system. Important decisions like task scheduling, resource management or partition schema should be tuned again almost every time one of these factors is modified.

Nowadays, there is no a universal strategy to configure a distributed data mining environment to be optimal in the resolution of every query. As an alternative, our contribution presents an appropriate support for multiple optimization and configuration strategies. Control policies may be changed, even while the system is running, to configure and optimize the performance of the system.
Although much effort has been addressed on the development of efficient distributed algorithms for data mining this is not the only way to outperform existing data mining solutions. Although the specific distributed implementation of a rule induction algorithm plays an important role, distribution schema, task scheduling and resource management are also key factors.
We consider that there is no global mechanism to evaluate all these factors before the actual system is running in a real environment. Resource management strategies, even data mining-oriented (Bestavros [4]), are too restrictive to provide an effective performance in multiple different circumstances.

Grid Computing services constitute a flexible manner of tackling the data mining needs. These services are described in next section.

4 Grid Computing

Grid Computing takes advantage of the low-load periods of all the computers connected to a network, making possible resource sharing. Grid environments differ substantially from conventional and parallel computing systems in their performance, cost, availability and security features. Different grid services are used with the aim of managing these characteristics. This section describes some of the most important grid services related to the DDM requirements, previously defined.

4.1 Job Scheduling and Planning

Schopf in [31] defines Grid Scheduling like a process involving resources over multiple administrative domains. The concept of scheduling in a grid environment involves the acquiring resources, the matching jobs to resources, the managing data and the monitoring progress of a job in the environment. In short, Grid Scheduling determines, reserves and allocates the resources of a Grid that are required for a job execution.

In order to do this, it is important to know the job requirements, such as the time assignment, required resources, data and software tools, network use and costs. By knowing these requirements, it is possible to plan a job. This process involves parsing the submitted job request, checking static information about available resources, pre-selecting resources, querying dynamic information of the resources, generating the schedule and delegating such job for its running.

The Grid Scheduling Architecture Research Group (GSA-RG) [12] of the GGF (Global Grid Forum) defines the Grid Scheduler Architecture (GSA). GSA allows cooperation between Local Resource Management Systems (LRMS) and other Grid Services such as Index Services and Monitoring and Discovery Services to facilitate the brokering decisions based on more knowledge about the system.

There are several schedulers with different capabilities. Some of the most importants are the following: PBS (Portable Batch System) [3], which controls the beginning of the scheduling of batch jobs and gets routing these jobs between different hosts; SGE (Sun Grid Engine) [13] provides batch queueing, load balancing, job statistics, user-specifiable resources and suspending and resuming jobs; Condor [7] supplies job queueing mechanism, scheduling policy, priority scheme and resource monitoring and management.

4.2 Data Grids

According to Chervenak et al. in [6] the access to distributed data is as important as access to distributed computational resources. Above all since many distributed scientific applications and, in our case, data mining applications require access to huge amounts of data. The volume of this analized data is measured in terabytes and petabytes.

One of the major goals of grid technology is to provide efficient access to data. Grids provide access to distributed computing and data resources, allowing data-intensive applications to improve significantly data access, management and analysis. Grid systems responsible for tackling and managing large amounts of data in geographically distributed environments are usually named data grids.

Data grids require the use of data sources, which are facilities that may accept or provide data [1]. These sources are composed of four components: (i) Storage systems, which include several file systems, caches, databases, and directory services, (ii) Data types, which include files, relational databases, XML databases and others, (iii) Data models, that is, different databases schemas, and (iv) Access mechanisms, which include file-system operations, SQL, XQuery, or XPath. Data grids must manage all these component in a dynamic fashion. Furthermore, generic data management systems involve more challenges. The most important ones are two:

1. All the layers of the grid infrastructure are characterized by their heterogeneity. This includes storage systems, computing systems, data access mechanisms and policies. Heterogeneity does not only affect to the infrastructure, but always to the data itself. Different kind of data formats and data from heterogeneous sources contribute to make more difficult an efficient data management.
2. This infrastructure must be able to tackle huge volumes of data, from terabytes to petabytes.

4.3 Access Restrictions and Control Policies

Foster defines in [8] a grid like “coordinate resources that aren’t subject to centralized control”. If there isn’t a centralized control and the resources are distributed in a wide area network crossing organizational boundaries, resources could be accessed by a lot of different organizations and it will be necessary to control the accesses to the resources. This is the reason why security is one of the most important aspects in the Grid technology.

In [9] the concept of Virtual Organization (VO) is defined and shows the boundaries between the VOs. It must be made sure that only certain organizations or users can

access certain resources, and especially that they are really who they claim that they are. In addition, the Grid Security Infrastructure (GSI) must transparently interact with common remote access tools, such as remote login, remote file access, and programming libraries like MPI.

GSI is based on public-key cryptography, and therefore can be configured to guarantee privacy, integrity, and authentication. But, it is advisable to take into account that, depending of the application, it may be interesting to minimize the cost of the security disabling the integrity and privacy features. Furthermore, GSI supports authorization in both the server-side and the client-side with different authorization options.

5 DMG: Data Mining Grid

Data mining is often used in commercial and business applications. There are significant examples of this scenario: data mining is used in the financial field for detecting several kind of frauds, or in entreprises applications with the aim of detecting trends and patterns in purchasing behaviors.

Respect to data grids, they are more adapted to scientific applications, where the total volume of data tends to be higher than that of business applications. Data mining applications also demand new alternatives in the field of discovery, data placement, scheduling, resource management, and transactional systems, among others. This is due in part to the following reasons:

- It is required to access to multiple databases and data holders, in general, because no single database is able to hold all data required by an application.
- In a generic scenario, multiple databases do not belong to the same institution and are not situated at the same location, but geographically distributed.
- For increasing the performance of some steps of the data mining process, it is possible to use local copies of the whole dataset or subsets.
- Business databases or datasets may be updated frequently, which implies replication and coherency problems.

Additionally, generic data management systems, and particularly data mining grids, involve a great number of challenges, defined in Section 4.2.

Current proposals do not address all these requirements through a general and complete solution. Thus, it is fundamental to define an infrastructure that provides basic and advanced data services for data mining applications, taking into account their peculiarities. Following these guidelines, we propose to define an infrastructure that include these components at least:

- Coupling data sources, which can be dynamically installed and configured. This characteristic makes easier data movement and replication. Nevertheless, the most important problem to deal with here is the potentially large size of datasets as well as the poor performance of the communications in a wide-area network. Data filtering, data replication and use of local datasets help to enhance the efficiency of data mining applications on grid infrastructures.

- Replicated data must be kept consistent. However, depending on the concrete application, this constraint may be relaxed, in such way that solving the consistency do not make worse the application performance.
- Authorized access to data resources, providing a controlled sharing of data within a virtual organization. This implies the use of authentication and access control mechanisms, in the form of access policies.
- One important feature in these systems is data discovery, which is a process that consists of discovering data based on metadata attributes. In this sense, publication, indexing and updating mechanisms are required.
- Both data preprocessing and analysis are basic in data mining applications. Related to these phases, planning and scheduling issues arise. Planning allows the application to be enhanced and adapted to existing resources. Scheduling is used for allocating these resources to the data mining tasks.

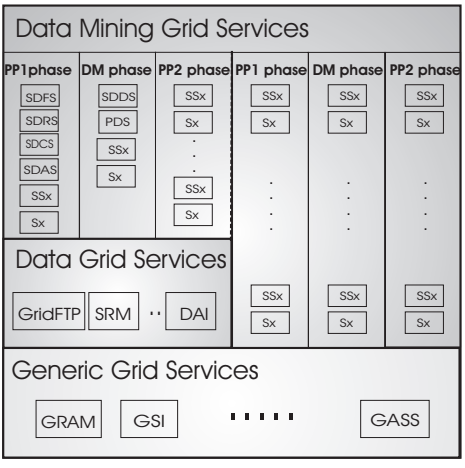
In [5], Cannataro et al. define the *Knowledge Grid* as an architecture built on top of a computational grid. This architecture extends the basic grid services with services of knowledge discovery on geographically distributed infrastructures. The Knowledge Grid services are organized in two layers: the core K-grid layer, which contains services directly implemented on top of basic grid services, and the high-level K-grid layer, which includes services used to describe and execute data mining processes.

Another different architecture has been proposed by Giannadakis et al. in [10], named *InfoGrid*. InfoGrid is mainly focused on the data integration. This infrastructure includes a layer of Information Integration Services, which enables heterogeneous information resources to be queried effectively. InfoGrid provides data integration within the framework of a data analysis process, and this is its major concern.

We propose a generic and vertical architecture based on the main data mining phases: pre-processing, data mining and post-processing (see Figure 1).

All these data mining services use both basic data and generic grid services. Since the deployment of every phase is independent of the rest, this architecture has as main advantage its flexibility. Furthermore, whenever a data or generic grid service can be used for a purpose, they will be used primarily. Only if the typical behavior of a service must be modified, a specialized service must be placed at the Data Mining Grid Services level, but implementing the same interface. Thus, data mining applications will become portable to other grid infrastructures. Specialization Services are shown in the figure. These are:

- *SDFS*, Specific Data Filtering Service: Due to the huge amount of data involved in the process of data mining, filtering data is an important task, solved by means of this service.
- *SDRS*, Specific Data Replication Service: One important aspect related to distributed data is data replication. This service is responsible for managing it.
- *SDCS*, Specific Data Consistency Service: Its main purpose is maintaining the data consistency in the grid.
- *SDAS*, Specific Data Access Service: This service is an adaptation of the DAI (Data Access and Integration) Data Grid Service to data mining applications on grid.
- *SDDS*, Specific Data Discovery Service: This service improves the discovery phase in the grid for mining applications.



GRAM: Grid Resource Allocation and Management
GSI: Grid Security Infrastructure
GASS: Grid Access to Secondary Storage
GridFTP: Grid File Transfer Protocol
SRM: Storage Resource Manager
DAI: Data Access and Integration
PP1 Phase: Pre-processing phase
DM Phase: Data mining phase
PP2 Phase: Post-processing phase
SDFS: Specific Data Filtering Service
SDRS: Specific Data Replication Service
SDCS: Specific Data Consistency Service
SDAS: Specific Data Access Service
SDDS: Specific Data Discovery Service
PDS: Patterns Detection Service
SSx: Specific Service x
Sx: Service x

Fig. 1. Proposed Data Mining Grid Architecture

– *SSx*: We can define additional specific services for the management of other features provided by the generic or data grid services. The architecture allows these services be included without changes in the rest of the framework.

New services oriented to data mining application are included in the architecture. For instance, *PDS* (Pattern Detection Service) is responsible for detecting data patterns. *Sx* represents additional services, which are not provided by basic grid services.

Another advantage of defining a three-tier architecture is the possibility to modify any one of the layers for any other technology. For example, it is possible to substitute the set of services of the pre-processing phase by CORBA services, if for instance our legacy system use this kind of technology. This way, different solutions can be integrated for the resolution of the process of Data Mining.

6 Conclusions and Future Work

This paper has shown differents ways of making easier the data mining process in a distributed environment. Many methods have been evaluated and we propose the Data Mining Grid as an enhanced solution to this problem. The main advantage of Data Mining

Grid is its adaptation to the requirements of an expensive data mining process. Firstly, the huge amount of data that is necessary to analize in a data mining application can be managed by means of data grids. Secondly, analized data is usually geographicaly distributed and belongs to different organizations. Thus, it is very important that the access policies are followed by every resource providers. Therefore, each provider could select the dataview shown to each user. Different views cause different results in data mining processes. Grid environments provides all the security infrastructure and the management of the access policies. Finally, one of the most important point in a data mining process is the data discovery and what actions it is necessary to do with each kind of data, since they can be in different formats. Brokering and sheduling systems implemented in Grids make possible the search of data and resources and the selection of actions adapted to each data format.

In order to perform the Data Mining Grid, we have defined a vertical architecture based on the main phases of data mining: pre-processing, data mining and post-processing. This architecture is flexible and allows all the services to be adapted to the data mining process, improving its performance.

As future work we aim to deploy a prototype to test our approach, by following the guidelines of our proposed architecture.

References

1. Malcolm Atkinson, Ann L. Chervenak, Peter Kunszt, Inderpal Narang, Norman W. Paton, Dave Pearson, Arie Shashoni, and Paul Watson. Data access, integration and management. *Chapter 22, The Grid: Blueprint for a New Computing Infrastructure, Second Edition*, pages 391–429, dec 2003.
2. Stuart Bailey, Emory Creel, Robert L. Grossman, Srinath Gutti, and Harimath Sivakumar. *Large-Scale Parallel Data Mining*, chapter A High Performance Implementation of the Data Space Transfer Protocol (DSTP), pages 55–64. 1999.
3. A. Bayucan, R. L. Henderson, C. Lesiak, B. Mann, T. Proett, and D. Tweten. PBS Portable Batch System. External Reference Specification.
4. Azer Bestavros. Middleware support for data mining and knowledge discovery in large-scale distributed information systems. In *Proceedings of the SIGMOD'96 Data Mining Workshop*, Montreal, Canada, June 1996.
5. Mario Cannataro and Domenico Talia. The knowledge grid. *Commun. ACM*, 46(1):89–93, 2003.
6. A. Chervenak, I. Foster, C. Kesselman, C. Salisbury, and S. Tuecke. The Data Grid: Towards an architecture for the distributed management and analysis of large scientific datasets. *Journal of Network and Computer Applications*, 23:187–200, 2001.
7. The Condor Project. *http://www.cs.wisc.edu/condor*.
8. I. Foster and C. Kesselman, editors. *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann, 1999.
9. I. Foster, C. Kesselman, and S. Tuecke. The Anatomy of the Grid: Enabling Scalable Virtual Organizations. *International Journal of SuperComputer Applications*, 15(3), 2001. Descripci n detallada de la arquitectura Grid.
10. N. Giannadakis, A. Rowe, M. Ghanem, and Y. Guo. InfoGrid: providing information integration for knowledge discovery. *Information Sciences. Special Issue: Knowledge Discovery from Distributed Information Sources*, 155(3–4):199– 226, October 2003.

11. Robert L. Grossman, Stuart M. Bailey, Harinath Sivakumar, and Andrei L. Turinsky. Papyrus: A system for data mining over local and wide-area clusters and super-clusters. In ACM, editor, *SC'99*. ACM Press and IEEE Computer Society Press, 1999.
12. Grid Scheduling Architecture Research Group (GSA-RG). *http://ds.e-technik.uni-dortmund.de/ yahya/ggf-sched/wg/arch-rg.htm*.
13. Grid Engine. *http://gridengine.sunsource.net*.
14. J. Huber. PPFS: An experimental filesystem for high performance parallel input/output. Master's thesis, Department of Computer Science, University of Illinois at Urbana-Champaing, 1995.
15. IBM. *Application programming interface and utility reference*. IBM DB2 Intelligent Miner for Data. IBM, September 1999.
16. ISL. Clementine servcer distributed architecture. White Paper, 1999. Integrates Solution Limited, SPSS Group.
17. Mahesh V. Joshi, Eui-Hong (Sam) Han, George Karypis, and Vipin Kumar. *CRPC Parallel Computing Handbook*, chapter Parallel Algorithms for Data Mining. Morgan Kaufmann, 2000.
18. C. Kamath and R. Musick. *Advanced in Distributed and Parallel Knowledge Discovery*, chapter Scalable data mining through fine grained parallesim: the present and the future, pages 29–77. AAAI Press / MIT Press, 2000.
19. H. Kargupta, W. Huang, S. Krishnamoorthy, and E. Johnson. Distributed clustering using collective principal component analysis. *Knoledge and Information Systems Journal Special Issue on Distributed and Parallel Knowledge Discovery*, 2000.
20. H. Kargupta, C. Kamath, and P. Chan. *Advanced in Distributed and Parallel Knowledge Discovery*, chapter Distributed and Parallel Data Mining: Emergence, Growth and Future Directions, pages 409–416. AAAI Press / MIT Press, 2000.
21. Hillol Kargupta and Philip Chan. *Advances in Distributed and Parallel Knowledge Discovery*, chapter Distributed and Parallel Data Mining: A Brief Introduction, pages xv–xxv. AAAI Press/MIT Press, 2000.
22. Hillol Kargupta, Ilker Hamzaoglu, and Brian Stafford. Scalable, distributed data mining-an agent architecture. page 211.
23. Kensingston, Enterprise Data Mining. Kensington: New generation enterprise data mining. White Paper, 1999. Parallel Computing Research Centre, Department of Computing Imperial College, (Contact Martin K hler).
24. Ron Kohavi. Data mining with mineset: What worked, what did not, and what might. In *Proceedings of the Workshop on Knowledge Discovery in Databases*, 1997. Workshop on the Commercial Success of Data Mining.
25. S. Krishnaswamy, S. W. Loke, and A. Zaslavsky. Cost models for distributed data mining. Technical Report 2000/59, School of Computer Science and Software Engineering, Monash University, Australia 3168, February 2000.
26. William A. Maniatty and Mohammed J. Zaki. A requirements analysis for parallel kdd systems. In Jose Rolim et al., editor, *3rd IPDPS Workshop on High Performance Data Mining*, pages 358–265, May 2000.
27. R. Musick. Supporting large-scale computational science. Technical Report UCRL–ID–129903, Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, Livermore, CA, 1998.
28. A. Prodromidis, P. Chan, and S. Stolfo. chapter Meta-learning in distributed data mining systems: Issues and approaches. AAAI/MIT Press, 2000.
29. Foster Provost. *Advances in Distributed and Parallel Knowledge Discovery*, chapter Distributed Data Mining: Scaling Up and Beyond, pages 3–28. AAAI Press/MIT Press, 2000.
30. Foster J. Provost and Venkateswarlu Kolluri. A survey of methods for scaling up inductive algorithms. *Data Mining and Knowledge Discovery*, 3(2):131–169, 1999.

31. J. M. Schopf. A General Architecture for Scheduling on the Grid. *Special Issue of JPDC on Grid Computing*, 2002.
Brokers y planificadores de tareas.
32. Colin Shearer. User driven data mining, 1996. Unicom Data Mining Conference. London.
33. T. Shintani and M. Kitsuregawa. Parallel algorithms for mining association rule mining on large scale PC cluster. In Zaki and Ho [37]. in conjunction with ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD99).
34. M. J. Zaki, C. T. Ho, and R. Agrawal. Parallel classification for data mining on shared-memory multiprocessors. In *Proceedings International Conference on Data Engineering*, March 1999.
35. Mohammed J. Zaki. *Scalable Data Mining for Rules*. PhD thesis, University of Rochester, July 1998. Published also as Technical Report #702.
36. Mohammed J. Zaki. Parallel sequence mining on SMP machines a data clustering algorithm on distributed memory machines a data clustering algorithm on distributed memory machines. In Zaki and Ho [37]. in conjunction with ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD99).
37. Mohammed J. Zaki and Ching-Tien Ho, editors. *Workshop on Large-Scale Parallel KDD Systems*, San Diego, CA, USA, August 1999. ACM. in conjunction with ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD99).
38. Mohammed J. Zaki and Ching-Tien Ho. Workshop report: Large-scale parallel KDD systems. In *SIGKDD Explorations* [37], pages 112–114. in conjunction with ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD99).