

Distributed Data Mining: Algorithms, Systems, and Applications

Byung-Hoon Park and Hillol Kargupta
Department of Computer Science and Electrical Engineering
University of Maryland Baltimore County
1000 Hilltop Circle Baltimore, MD 21250

Introduction

Advances in computing and communication over wired and wireless networks have resulted in many pervasive distributed computing environments. The Internet, intranets, local area networks, and wireless networks are some examples. Many of these environments have different distributed sources of voluminous data and multiple compute nodes. Analyzing and monitoring these distributed data sources require data mining technology designed for distributed applications.

This chapter starts by pointing out a mismatch between the architecture of most off-the-shelf data mining systems and the needs of mining systems for distributed applications. It also claims that such mismatch may cause a fundamental bottleneck in many emerging distributed applications. Figure 1(Left) presents a schematic diagram of the traditional data warehouse-based architecture for data mining. This model of data mining works by regularly uploading mission critical data in the warehouse for subsequent centralized data mining application. This centralized approach is fundamentally inappropriate for most of the distributed and ubiquitous data mining applications. The long response time, lack of proper use of distributed resources, and the fundamental characteristics of centralized data mining algorithms do not work well in distributed environments.

A scalable solution for distributed applications calls for distributed processing of data, controlled by the available resources and human factors. For example, consider an ad hoc wireless sensor network where the different sensor nodes are monitoring some time-critical events. Central collection of data from every sensor node may create heavy traffic over the limited bandwidth wireless channels and this may also drain a lot of power from the devices. A distributed architecture for data mining is likely to reduce the communication load and also reduce the battery power more evenly across the different nodes in the sensor network. One can easily imagine similar needs for distributed computation of data mining primitives in ad hoc wireless networks of mobile devices like PDAs, cellphones, and wearable computers. Potential applications include personalization, collaborative process monitoring, intrusion detection over ad hoc wireless networks. We need data mining architectures that pay careful attention to the distributed resources of data, computing, and communication in order to consume them in a near optimal fashion. Distributed data mining (DDM) considers data mining in this broader context. As shown in Figure 1 (Right), the objective of DDM is

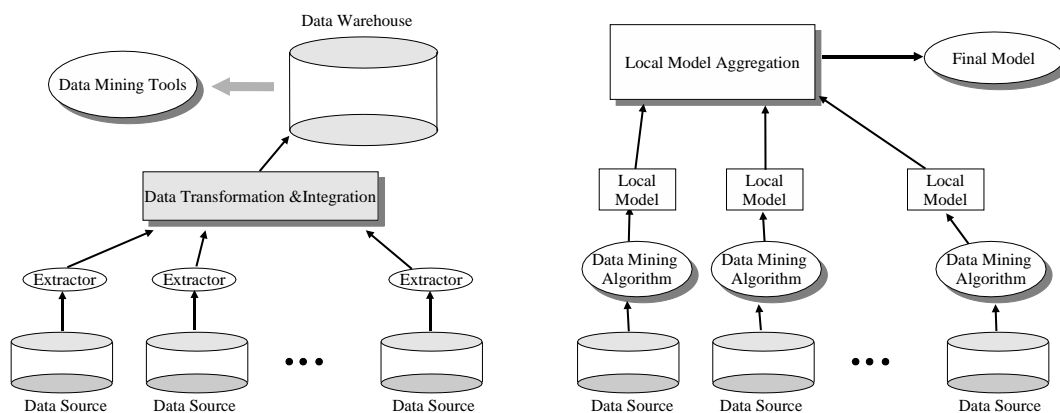


Figure 1. A data warehouse architecture (Left). Distributed Data Mining Framework (Right).

to perform the data mining operations based on the type and availability of the distributed resources. It may choose to download the data sets to a single site and perform the data mining operations at a central location. However, that decision in DDM should be based on the properties of the computing, storage, and communication capabilities. This is in contrast with the traditional centralized data mining methodology where collection of data at a single location prior to analysis is an invariant characteristic.

The wireless domain is not the only example. In fact, most of the applications that deal with time-critical distributed data are likely to benefit by paying careful attention to the distributed resources for computation, storage, and the cost of communication. The world wide web is a very good example. It contains distributed data and computing resources. An increasing number of databases (e.g. weather databases, oceanographic data at www.noaa.gov), and data streams (e.g. financial data at www.nasdaq.com, emerging disease information at www.cdc.gov) are coming online; many of them change frequently. It is easy to think of many applications that require regular monitoring of these diverse and distributed sources of data. A distributed approach to analyze this data is likely to be more scalable and practical particularly when the application involves a large number of data sites. The distributed approach may also find applications in mining remote sensing and astronomy data. For example, the NASA Earth Observing System (EOS), a data collector for a number of satellites, holds 1450 data sets that are stored, managed, and distributed by the different EOS Data and Information System (EOSDIS) sites that are geographically located all over the USA. A pair of Terra spacecraft and Landsat 7 alone produces about 350 GB of EOSDIS data per day. An online mining system for EOS data streams may not scale if we use a centralized data mining architecture. Mining the distributed EOS repositories and associating the information with other existing environmental databases may benefit from DDM. In astronomy, the size of telescope image archives have already reached the terabyte range and they continue to increase very fast as information is collected for new all-sky surveyors such as the GSC-II (McLean et al., 1998) and the Sloan Digital Survey (Szalay, 1998). DDM may offer a practical scalable solution for mining these large distributed astronomy data repositories.

DDM may also be useful in environments with multiple compute nodes connected over high speed networks. Even if the data can be quickly centralized using the relatively fast network, proper balancing of computational load among a cluster of nodes may require a distributed approach. The privacy issue is playing an increasingly important role in the emerging data mining applications. If a consortium of different banks wants to collaborate for detecting frauds then a centralized data mining system may require collection of all the data from every bank in a single location. However, this is not necessarily true if DDM is our choice of technology. DDM systems may be able to learn models from distributed data without exchanging the raw data. This may allow both detection of fraud and preserving the privacy of every Bank's customer transaction data.

This paper presents a brief overview of the DDM algorithms, systems, applications, and the emerging research directions. The structure of the paper is organized as follows. We first present the related research of DDM and illustrate data distribution scenarios. Then DDM algorithms are reviewed. Subsequently, the architectural issues in DDM systems and future directions are discussed.

Related Research

DDM deals with distributed data analysis algorithms and distributed systems. There are several other fields that deal with these issues at least partially. This section discusses this connection between DDM and a few other related fields.

Many DDM systems adopt the Multi-Agent System (MAS) architecture. MAS finds its root in the Distributed Artificial Intelligence (DAI), which investigates AI-based search, learning, planning and other problem-solving techniques for distributed environments. Early research in this area includes blackboard systems (Nii, 1986), classifier systems (Holland, 1975), production systems (Newell & Simon, 1963), connectionism (Rumelhart & McClelland, 1986), Minsky's Society of Mind concept (Minsky, 1985), Cooperative problem solving (Durfee, Lesser, & Corkill, 1989), Actor framework (Agha, 1986), the Contract Net protocol (Smith, 1980; Davies & Smith, 1983). The emergence of distributed environments, such as the Internet and e-commerce have catalyzed many applications of DAI/MAS technology and extensive literature on multi-agent communication (Finin, Labrou, & Mayfield, 1997), negotiation (Rosenschein, 1994), search (Lander & Lesser, 1992), architectural issues (Woolridge & Jenneings, 1995), and learning (Sen, 1997) is now available. While most of these topics are quite relevant to the DDM, DAI/MAS learning and architectural issues are probably the most relevant topics. The existing literature on multi-agent learning does not typically address the issues involved with large scale distributed data analysis. In DAI/MAS the focus is more on learning control knowledge (Byrne & Edwards, 1995; Carmel & Markovitch, 1995; Joshi, 1995; Sen & Sekaran, 1995), adaptive behavior (Mor, Goldman, & Rosenschein, 1995; Sandholm & Crites, 1995; Weiß, 1995), and other related issues. However, several efforts reported in the DAI/MAS literature do consider data intensive applications such as information discovery in the World Wide Web (Lesser et al., 1998; Menczer & Belew, 1998; Moukas, 1996).

High performance parallel computing environments are often used for quick access and manipulation of such data sets. Therefore, it makes sense to exploit such computing environments for scaling up the data mining process. Parallel data mining (PDM) (Alsabti, Ranka, & Singh, 1997; Freitas & Lavington, 1998; Kamath & Musick, 2000; Zaki, 1996,

1997; Parthasarathy, Zaki, Ogihara, & Li, 2001; Han, Karypis, & Kumar, 1997; Joshi, Han, Karypis, & Kumar, 2000) does this. Although PDM often assumes the presence of high speed network connections among the computing nodes, the development of DDM has also been influenced by the PDM literature.

Data fusion refers to seamless integration of data from disparate sources. Among the extensive literature on data fusion, the distributed approach of multi-sensor data fusion is worth considering. Within this approach, each sensor makes a local decision. All local decisions are then combined at a fusion center to produce a global decision. The objective of this approach is to determine the optimum local and global decision rules that maximize the probability of signal detection. Typically the decision requires hypothesis testing techniques. The Bayesian (Hoballah & Varshney, 1989) and the Neyman-Pearson (Viswanathan & Varshney, 1997) criteria are often used for this purpose. The following section presents data distribution scenarios and steps required to prepare data for DDM.

Data Distribution and Pre-Processing

Identifying how the data is distributed is the first step in developing a distributed data mining solution. Most of the DDM algorithms are designed for the relational data model (tabular form). That is why in this chapter we shall restrict our attention to the relational model and discuss different data distribution scenarios within the context of the data schema.

Homogeneous/Heterogeneous Data Scenarios

In a relational database the schema provides the information regarding the relations stored. Information regarding different schemata from different tables is essential for identifying their mutual dependencies and therefore the choice of data mining algorithms. Most of the existing DDM work considers homogeneous schemata across different sites. Homogeneous schemata contain the same set of attributes across distributed data sites. This distributed data model usually occurs in the same organization (e.g, Wal-Mart chains) or across similar domains. Some DDM algorithms consider heterogeneous schemata that define different sets of attributes across distributed databases. However, the heterogeneous schemata are usually restricted to a simple scenario where every participating table shares a common key column that links corresponding rows across the tables.

Preparing the data is an important step in data mining and DDM is no exception. Data pre-processing in DDM must work in a distributed fashion. Many of the standard centralized data pre-processing techniques can be directly applied without downloading all the data sets to a single site. Some of these techniques are briefly discussed in the following section.

Data Pre-processing

Standardizing data sets across different sites is an important process in DDM. The first step is to exchange the database schema information and the meta-data. Typically this involves low communication overhead. Additional information regarding the physical meaning of features, measurement units, and other domain specific information are often exchanged for better understanding of the distributed data sources.

If the data sites are heterogeneous, key-Association is a necessary step. The primary purpose of the key-Association step is to select a set of keys for associating the data across different sites. The schema information and the physical meaning of the features can be used for linking the distributed data sets. If a precise key is not available we may need to use clustering techniques to create approximate keys. Usually in a spatial database the coordinate location can serve as a key. In a temporal database the time stamp of the observation may serve the same purpose.

Data normalizations are often necessary for avoiding undesirable scaling effects. The need for data normalization depends on the application and the data analysis algorithm. For example, normalization may be critical in nearest neighbor classification for assigning uniform weight to all the features. Some of the popular normalization techniques are:

1. Decimal scaling: This technique works by moving the decimal point. It is also applicable to both homogeneous and heterogeneous DDM.

2. Standard deviation scaling: For any given feature value $x[i]$ this technique constructs the normalized feature $x'[i]$ where $x'[i] = \frac{x[i] - \mu_x}{\sigma_x}$; μ_x and σ_x are the mean and the standard deviation of $x[i]$. In case of homogeneous DDM, the computation of overall mean and standard deviation can be distributed among the different sites in a straight forward manner. In case of heterogeneous DDM this computation is strictly local.

Missing data is a real problem in most applications. Most of the simple techniques like replacement by (1) class labels, (2) some constant value, and (3) expected value may or may not work depending on the application domain. Usually they bias the data set and may result in poor data mining performance. However, if desired these techniques can be used directly in a DDM application. More involved techniques for handling missing data require predictive modeling of data. Typically decision trees, Bayesian algorithms, and other inductive models are learned for predicting the missing values. The following section considers related fields of DDM.

Distributed Data Mining Algorithms

Most DDM algorithms are designed upon the potential parallelism they can apply over the given distributed data. Typically the same algorithm operates on each distributed data site concurrently, producing one local model per site. Subsequently all local models are aggregated to produce the final model. In essence, the success of DDM algorithms lies in the aggregation. Each local model represents locally coherent patterns, but lacks details that may be required to induce globally meaningful knowledge. For this reason, many DDM algorithms require a centralization of a subset of local data to compensate it. Therefore, minimum data transfer is another key attribute of the successful DDM algorithm. In this section, we present a literature review on DDM algorithms.

Distributed Classifier Learning

Most distributed classifiers have their foundations in ensemble learning (Dietterich, 2000; Opitz & Maclin, 1999; Bauer & Kohavi, 1999; Merz & Pazzani, 1999). The ensemble approach has been applied in various domains to increase the classification accuracy of predictive models. It produces multiple models (base classifiers) — typically from “homogeneous” data subsets — and combines them to enhance accuracy. Typically, voting

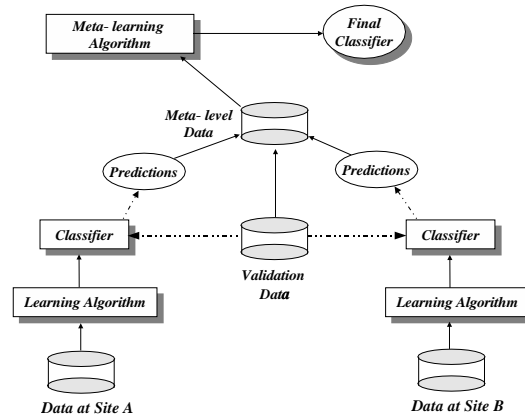


Figure 2. Meta Learning from distributed homogeneous data sites.

(weighted or unweighted) schemes are employed to aggregate base classifiers.

The ensemble approach is directly applicable to the distributed scenario. Different models can be generated at different sites and ultimately aggregated using ensemble combining strategies. Fan, et al. (Fan, Stolfo, & Zhang, 1999) discussed an AdaBoost-based ensemble approach in this perspective. Breiman (Breiman, 1999) considered Arcing as a mean to aggregate multiple blocks of data, especially in on-line setting. An experimental investigation of Stacking (Wolpert, 1992) for combining multiple models was reported elsewhere (Ting & Low, 1997).

Homogeneous Distributed Classifiers. One notable ensemble approach to learn distributed classifier is meta-learning framework (Chan & Stolfo, 1993b, 1993a, 1998). It offers a way to mine classifiers from homogeneous, distributed data. In this approach, supervised learning techniques are first used to learn classifiers at local data sites; then meta-level classifiers are learned from a data set generated using the locally learned concepts. The meta-level learning may be applied recursively, producing a hierarchy of meta-classifiers. Java Agent for Meta-learning is reported elsewhere (Stolfo et al., 1997; Lee, Stolfo, & Mok, 1999). Meta-learning follows three main steps:

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

Learning at the meta-level can work in many different ways. For example, we may generate a new dataset using the locally learned classifiers. We may also move some of the original training data from the local sites, blend it with the data artificially generated by the local classifiers, and then run any learning algorithm to learn the meta-level classifiers. We may also decide the output of the meta-classifier by counting votes cast by different base classifiers. The following discourse notes two common techniques for meta-learning from the output of the base classifiers are briefly described in the following.

1. The Arbiter Scheme: This scheme makes use of a special classifier, called arbiter,

for deciding the final class prediction for a given feature vector. The arbiter is learned using a learning algorithm. Classification is performed based on the class predicted by the majority of the base classifiers and the arbiter. If there is a tie, the arbiter's prediction gets the preference.

2. The Combiner Scheme: This combiner scheme offers an alternate way to perform meta-learning. The combiner classifier is learned in either of the following ways. One way is to learn the combiner from the correct classification and the base classifier outputs. Another possibility is to learn the combiner from the data comprised of the feature vector of the training examples, the correct classifications, and the base classifier outputs. Either of the above two techniques can be iteratively used resulting in a hierarchy of meta-classifiers. Figure 2 shows the overall architecture of the meta learning framework.

Meta-learning illustrates two characteristics of DDM algorithms — parallelism and reduced communication. All base classifiers are generated in parallel and collected at the central location along with the validation set, where the communication overhead is negligible compared to the transfer of entire raw data.

Distributed Learning with Knowledge Probing (DLKP) (Guo & Sutiwaraphun, 2000) is another meta-learning based technique to produce a global model by aggregating local models. Knowledge probing was initially proposed to extract descriptive knowledge from a black box model, such as neural network. The key idea is to probe a descriptive model from data whose class values are assigned by a black box model. DLKP is an extension of knowledge probing to a homogeneous distributed data setting. It works as follows:

1. Generate base classifiers at each site using off-the-shelf classifier learning algorithms.
2. Select a set of unlabeled data for the probing set.
3. Prepare probing data set by combining predictions from all base classifiers.
4. Learn a final model directly from the probing set.

In step 3, a probing data set can be generated using various methods such as uniform voting, trained predictor, likelihood combination, etc. The main difference between meta-learning and DLKP is the second learning phase. In meta-learning, special type of classifiers (meta-classifier) are trained to combine or arbitrate the outputs of the local models. The final classifier includes both meta-classifiers and local (base) models. In contrast, DLKP produces a final descriptive model that is learned from the probing data set as its final classifier.

Gorodetski and his colleagues (Gorodetski, Skormin, Popyack, & Karsaev, 2000) addressed distributed learning in data fusion systems within the meta-learning paradigm. For *base classifiers*, they developed a technique that learns a wide class of rules from arbitrary formulas of first order logic. This is particularly applied as a visual technique to learn rules from databases. To overcome deficiencies of local learning (base classifiers), they adopted a randomized approach to select subsets of attributes and cases that are required to learn rules from distributed data, which results in a meta-level classifier.

Heterogeneous Distributed Classifiers. The ensemble learning based approach offers techniques for mining from homogeneous data sites. However, it is not straightforward to apply to heterogeneous distributed data. In heterogeneous distributed data, we observe the incomplete knowledge about the complete data set. Different local models represent

disjoint regions of the problem and DDM has to develop a global data model, associations, and other patterns with only limited access to the features observed at non-local sites. For this reason, it is generally believed that mining of heterogeneous distributed data is more challenging.

The issues in mining from heterogeneous data is discussed in (Provost & Buchanan, 1995) from the perspective of inductive bias. This work notes that such heterogeneous partitioning of the feature space can be addressed by decomposing the problem into smaller sub-problems when the problem is site-wise decomposable. However, this approach is too restrictive to handle problems that involve inter-site correlations.

The WoRLD system (Aronis, Kulluri, Provost, & Buchanan, 1997) addressed the problem of concept learning from heterogeneous sites by developing an “activation spreading” approach. This approach first computes the cardinal distribution of the feature values in the individual data sets. Next, this distribution information is propagated across different sites. Features with strong correlations to the concept space are identified based on the first order statistics of the cardinal distribution. Since the technique is based on the first order statistical approximation of the underlying distribution, it may not be appropriate for data mining problems where concept learning requires higher order statistics.

An ensemble approach to combine heterogeneous local classifiers is proposed in (Tumer & Ghosh, 2000). It especially uses an order statistics-based technique for combining high variance models generated from heterogeneous sites. The technique works by ordering the predictions of different classifiers and using them in an appropriate manner. The paper gives several methods, including selecting an appropriate order statistic as the classifier and taking a linear combination of some of the order statistics (“spread” and “trimmed mean” classifiers). It also analyzes the error of such a classifier in various situations. Although these techniques are more robust than other ensemble based models, they do not consider global correlations.

Park and his colleagues (Park et al., 2002) note that any inter-site pattern cannot be captured by the aggregation of heterogeneous local classifiers. To detect such patterns, they first identify a subset of data that any local classifier can not classify with a high confidence. Identified subset is merged in a central site and another classifier (central classifier) is constructed from it. When a combination of local classifiers can not classify an unseen data with a high confidence, the central classifier is used instead. This approach exhibits a better performance than a simple aggregation of local models. However, its performance is sensitive to the sample size (or, confidence threshold).

Collective Data Mining

Kargupta and his colleagues considered the *Collective* framework to address data analysis for heterogeneous environments and proposed the *Collective Data Mining* (CDM) framework for predictive data modeling. CDM is a functionally complete framework for inducing any pattern function in a distributed fashion that has roots in theory of communications, machine learning, statistics, and distributed databases. Instead of combining incomplete local models, it seeks to find globally meaningful pieces of information from each local site. In other words, it obtains local building blocks that directly constitute the global model. Given a set of labeled training data, CDM learns a function that approximates it. The foundation of CDM is based on the observation that any function can be represented in

a distributed fashion using an appropriate set of basis functions. When the basis functions are orthonormal, the local analysis produce correct and useful results that can be directly used as a component of the global model without any loss of accuracy. Since data modeling using canonical, non-orthogonal basis functions does not appear to be suitable in a distributed environment, CDM does not directly learn data models in popular representations like polynomial, logistic functions, decision tree, and feed-forward neural-nets. Instead, it first learns the spectrum of these models in some appropriately chosen orthonormal basis space, guarantees the correctness of the generated model, and then converts the model in orthonormal representation to the desired forms. The main steps of CDM can be summarized as follows:

1. Generate approximate orthonormal basis coefficients at each local site;
2. Move an appropriately chosen sample of the data sets from each site to a single site and generate the approximate basis coefficients corresponding to non-linear cross terms;
3. Combine the local models, transform the model into the user described canonical representation, and output the model.

Here non-linear terms represent a set of coefficients (or patterns) that can not be determined at a local site. In essence, the performance of a CDM model depends on the quality of estimated cross-terms. Typically CDM requires an exchange of a small sample that is often negligible compared to entire data. For example, let us consider the Collective Principal Component Analysis (CPCA) algorithm (Kargupta, Huang, Krishnamrthy, Park, & Wang, 2000; Kargupta, Huang, S., & Johnson, 2001) that performs distributed PCA. The followings are main steps of CPCA.

1. Perform local PCA at each site; select dominant eigenvectors and project the data along them.
2. Send a sample of the projected data along with the eigenvectors.
3. Combine the projected data from all the sites.
4. Perform PCA on the global data set, identify the dominant eigenvectors and transform them back to the original space.

To compute exact Principal Components (PCs), in principal, we need to reconstruct the original data from all projected local samples. However, since the PCA is invariant to linear transformation, the global PCs are computed directly from projected samples. The size of samples is a lot smaller than that of the original data. In other words, we can exploit the dimensionality reduction already achieved at each of the local sites.

Kargupta, et. al. (Kargupta et al., 2001) proposed a distributed clustering algorithm based on CPCA. The proposed work first apply the given off-the-shelve clustering algorithm to the local PCs. Then the global PCs are obtained from an appropriate data subset (projected) that is union of all representative points from local clusters. Each site projects local data on the global PCs and again obtain new clusters, which are subsequently combined at the central site.

CDM also shows that with an appropriate basis, multivariate polynomial regression can be performed from heterogeneous distributed data. The collective multivariate regression (Hershberger & Kargupta, 2001) chooses wavelet basis to represent local data. For each feature in data, wavelet transformation is applied and significant coefficients are centralized. Then the regression is performed directly on the wavelet coefficients. This approach has a significant advantage in communication reduction since a set of wavelet coefficients usually

represents raw data in a highly compressed format.

The CDM framework is also extended to other areas like Bayesian Network (BN) learning (Chen, Krishnamoorthy, & Kargupta, 2001, 2002). Within collective Bayesian network learning strategy, each site compute a BN and identifies the observations that are most likely to be evidence of coupling between local and non-local variables. These observations are used to compute a non-local BN consisting of links between variables across two or more sites. The final collective BN is obtained by combining the local models with the links discovered at the central site.

Other extensions of the CDM framework include distributed decision tree construction (Park, Ayyagari, & Kargupta, 2001), and collective hierarchical clustering (Johnson & Kargupta, 1999).

Distributed Association Rule Mining

Two main approaches to distributed association rule mining are Count Distribution (CD) and Data Distribution (DD). CD especially considers the case when the data is partitioned homogeneously into several data sites. Each data site computes support counts for the same candidate itemsets independently, which are then gathered at a central site to determine the large itemsets for the next round. In contrast, DD focuses on maximizing parallelism; it distributes candidate itemsets so that each site computes a disjoint subset. It requires the exchange of data partitions, therefore only viable for machines with high speed communications.

Agrawal and Shafer (Agrawal & Shafer, 1996) introduce a parallel version of Apriori. It requires $O(|C| \cdot n)$ communication overhead for each phase, where $|C|$ and n are the size of candidate itemset C and the number of data sites, respectively. The Fast Distributed Mining (FDM) algorithm (Cheung, Ng, Fu, & Fu, 1996) reduces the communication cost to $O(|C_p| \cdot n)$, where C_p is the potential candidate itemset (or, the union of all locally large itemsets). The FDM notes that any globally large itemset should be identified locally large at one or more sites. However, this approach does not scale well in n , especially when the distributed data are skewed in distribution. Schuster and Wolff (Schuster & Wolff, 2001) propose the Distributed Decision Miner (DDM) algorithm that reduces communication overhead to $O(Pr_{above} \cdot |C| \cdot n)$, where Pr_{above} is the probability that a candidate itemset has support greater than the given threshold. DDM differs from FDM in that a locally large itemset is not identified a globally large itemset until it is verified by exchange of messages.

Jensen and Sopakar (Jensen & Soparkar, 2000) propose an association rule mining algorithm from heterogeneous relational tables. It particularly considers mining from star schema of n primary tables T_1, \dots, T_n (with one primary key) and one relationship table T_r . They assume T_r contains all foreign keys to each T_i , and exploit the foreign key relationships to develop a decentralized algorithm. Since each foreign key is a unique primary key in a corresponding table, explicit join operation can be avoided to compute support of an itemset.

Distributed Clustering

Most distributed clustering algorithms have their foundations in parallel computing, and are thus applicable in homogeneous scenarios. They focus on applying center-based clustering algorithms, such as K-Means, K-Harmonic Means and EM, in a parallel fashion (Dhillon & Modha, 1999; Zhang, Hsu, & Forman, 2000; Sayal & Scheuermann, 2000). Two

approaches exist in this category. The first approach approximates the underlying distance measure by aggregation and the second provides the exact measure by data broadcasting. The approximation approach is sensitive to aggregation ratio and the exact approach involves heavy communication overheads.

Forman and Zhang (Forman & Zhang, 2000) propose a center-based distributed clustering algorithm that only requires the exchange of sufficient statistics, which is essentially an extension of their earlier parallel clustering work (Zhang et al., 2000). The Recursive Agglomeration of Clustering Hierarchies by Encircling Tactic (RACHET) (Samatova, Ostrouchov, Geist, & Melechko, 2002) is also based on the exchange of sufficient statistics. It particularly collects local dendograms that are merged into a global dendogram. Each local dendogram contains descriptive statistics about the local cluster centroid that is sufficient for the global aggregation. However, both approaches need to iterate until the sufficient statistics converge or the desired quality is achieved.

Parthasarathy and Ogihara (Parthasarathy & Ogihara, 2000) note that the primary problem with distributed clustering is to provide a suitable distance metric. They define one such metric as based on the association rule. However, this approach is still restricted to homogeneous tables. In contrast, McClean and her colleagues (McClean, Scotney, & Greer, 2000) consider the clustering of heterogeneous distributed databases. They particularly focus on clustering heterogeneous datacubes comprised of attributes from different domains. They utilize Euclidean distance and Kullback-Leiber information divergence to measure differences between aggregates.

The PADMA system (Kargupta, Hamzaoglu, Stafford, Hanagandi, & Buescher, 1996; Kargupta, Hamzaoglu, & Stafford, 1997) is an application system that employs a distributed clustering algorithm. It is a document analysis tool from homogeneous data sites, where clustering is aided by relevance feedback-based supervised learning techniques.

Other DDM algorithms

A distributed cooperative Bayesian learning algorithm was developed in (Yamanishi, 1997). This technique considers homogeneous data sets. In this approach different Bayesian agents estimate the parameters of the target distribution, and a population learner combines the outputs of those Bayesian models. A “fragmented approach” to mine classifiers from distributed data sources is suggested by (Cho & Wüthrich, April, 1998). In this method, a single, good, rule is generated in each distributed data source. These rules are then ranked using some criterion and a number of the top ranked rules are selected to form the rule set. In (Lam & Segre, 1997) the authors report a technique to automatically produce a Bayesian belief network from knowledge discovered using a distributed approach. Additional work on DDM design optimization (Turinsky & Grossman, 2000), classifier pruning (Prodromidis & Stolfo, 2000), measuring the quality of distributed data sources (Wüthrich, Cho, Pun, & Zhang, 2000), and problem decomposition and local model selection in DDM (Pokrajac, Fiez, Obradovic, Kwek, & Obradovic, 1999), are also reported.

Distributed Data Mining Systems

A DDM system is inevitably a very complex entity that is comprised of many components; mining algorithms, communication subsystem, resource management, task scheduling, user interfaces, etc. It should provide efficient access to both distributed data and

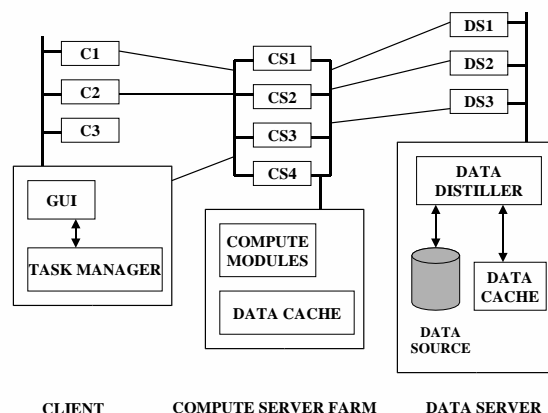


Figure 3. Intelliminer DDM system. C1,C2 and C3 are clients. CS1, CS2 and CS3 are computing servers. DS1, DS2 and DS3 are data servers

computing resources, monitor the entire mining procedure, and present results to users in appropriate formats. A successful DDM system is also flexible enough to adapt to various situations. It should dynamically identify the optimal mining strategy under the given resources and provide an easy way to update its components. In this section, we discuss various aspects of DDM systems. In particular, architectural and communication issues are examined.

Architectural Issues

Many organizations have a cluster of high-performance workstations (e.g, SMPs) connected by a network link. Such a cluster can be a cost effective resource for scalable data mining. However, Parthasarathy (Parthasarathy, 2001) notes that performance in such an environment is largely affected by contention for processors, network link and I/O resources. He emphasized that TCP/IP protocol is inherently designed to avoid contention, thus reducing communication rates drastically even with a small amount of resource competition. As an approach to deal with such a problem, he suggests guarding the allocation of resources and making applications adapt to resource constraints. The *Three-tier client/server* architecture is one approach for efficient resource management. The Kensington system (Chatratchat et al., 1999), Intelliminer (Parthasarathy & Subramonian, 2000) belong to this category. For example, the Kensington system is divided into client, application server and third-tier server. The client module provides interactive creation of data mining tasks, visualization of data and models and sampled data, while the application server is responsible for user authentication, access control, task coordination and data management. The third-tier server provides high performance data mining services located in high-end computing facilities that include parallel systems. Particularly, it is placed in proximity to the databases to increase the performance. Figure 3 shows the Intelliminer developed by Parthasarathy and Subramonian (Parthasarathy & Subramonian, 2000). Intelliminer is most specifically designed to support distributed *doall loop* primitive over clusters of SMP workstations. The *doall loop* is one where each iteration is independent (Wolfe, 1995).

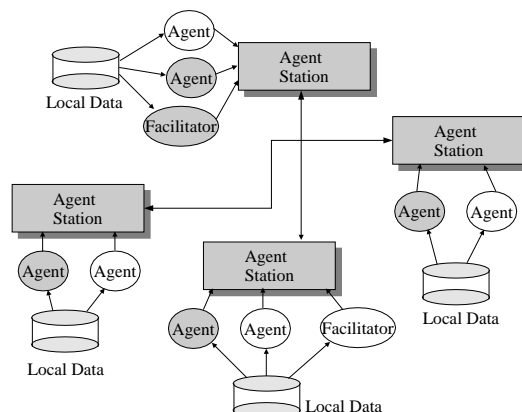


Figure 4. BODHI: An Agent-based DDM architecture.

The agent-based model is another approach to address scalable mining over distributed data of large sizes (See figure 4). Although there are many different types of software agents (Wooldridge & Jennings, 1995), they are typically considered to be autonomous intelligent softwares. All of the agent-based DDM systems employ one or more agents per data site. These agents are responsible for analyzing local data and communicate with other agents during the mining stage. A globally coherent knowledge is synthesized via exchanges of locally mined knowledge. However, in an agent-based model, an efficient control over remote resources is inherently difficult. In addition, without dedicated high-performance compute servers, the optimal mining performance is often not guaranteed. For this reason, most agent-based systems require a supervisory agent that facilitates the entire mining process. This agent, sometime called a *facilitator*, controls the behavior of each local agent. Java Agents for Meta-learning (JAM) and the BODHI system follow this approach. In JAM, for example, agents operating on a local database produce local classifiers. These local classifiers are then imported to a data site where they are combined using meta-learning agents. Both BODHI and JAM are implemented with Java, thus realizing a platform independent distributed data mining environment. By adopting loosely synchronized communication protocol among data sites, they seek to achieve asynchronous distributed data mining. BODHI also notes the importance of mobile agent technology. As all agents are extensions of a basic agent object, the BODHI system is easily capable of transferring an agent from one site to another, along with the agent's environment, configuration, current state, and learned knowledge.

The InfoSleuth project (Martin, Unruh, & Urban, 1999) at the Microelectronics and Computer Corporation (MCC) is agent technology-based distributed infrastructure. It implements various types of agents that facilitate information gathering, and analysis and event notification. The InfoSleuth is designed to provide an integrated solution to information-related problems that are tailored for user-specified knowledge discovery. It especially supports continuous query subscriptions that instruct a targeted agent to continuously monitor and update its response to the query if any change is detected from an information source.

The most notable aspect of the InfoSleuth system is its support of composite event detection. Events in the InfoSleuth define both the changes and analysis observations of data that are transferred from various heterogeneous data sources. Data change events and data analysis events are combined to create higher-level events called composite events. InfoSleuth provides the composite event language that is adopted from active database rule language.

Distributed Knowledge Networks (DKN) (Honavar, Miller, & Wong, 1998) is another distributed data mining framework that is based on agent technology. DKN emphasizes the important role of mobile agents in situations where the transfer of data from private source becomes infeasible. DKN also proposes a detailed solution to extract and integrate data from heterogeneous sources that consist of different data types and semantic structures. To facilitate interoperability among heterogeneous databases, it adopts an object-oriented view that creates a uniform interface for multidatabases. An object-oriented view helps to hide the heterogeneity and distributed nature of multidatabases. Rooted in knowledge-based agent software, DKN also implements an object-oriented data warehouse.

Communication Models in DDM Systems

Architectural requirements for efficient data communication for a wide area network are explored in (Grossman et al., 1998; Turinsky & Grossman, 2000), and most often consider data transfer costs in finding an optimal mining process. In a distributed data/compute nodes environment, they formulate the overall cost function for data transfer strategies. A strategy denotes the amount of data transfer between every pair of compute nodes within the environment. The optimal strategy is one that minimizes the cost function with respect to the given error level. The problem is essentially a convex linear programming problem, and the OPTimal strategy Data and Model Partition strategy (OPTDMP) is proposed as a solution. However, their work is restricted to identify the amount of data to transfer. It does not address any specific portion of data to be transferred, and thus is inapplicable to the heterogeneous case.

Papyrus (Grossman, Bailey, Sivakumar, & Turinsky, 1999; Grossman et al., 1999) is designed to find the optimal mining strategy over clusters of workstations connected by either a high-speed network (super-clusters) or a commodity network (meta-clusters). One such environment is shown in Figure 5. Papyrus supports three strategies: Move Results (MR), Move Models (MM) and Move Data (MD), as well as combinations of these strategies.

Costing infrastructure is also discussed in the DDM architecture by Krishnaswamy, et. al. (Krishnaswamy, Zaslavsky, & Loke, 2000). He notes that DDM evolves to embrace the e-commerce environment, especially the paradigm of Application Service Providers (ASP). ASP technology allows small organizations or individuals to access a pool of commercial software on demand (Sarawagi & Nagaralu, 2000). The proposed architecture demonstrates how DDM can be integrated into ASP in an e-commerce environment. Krishnaswamy emphasizes that the primary issue under such highly inter-domain operational environments is how to set up a standard by which to bill each user based on estimated costs and response times.

Components Maintenance

Expandability of components is one key feature of the successful DDM systems. There are far too many different approaches and algorithms for data mining to incorporate them

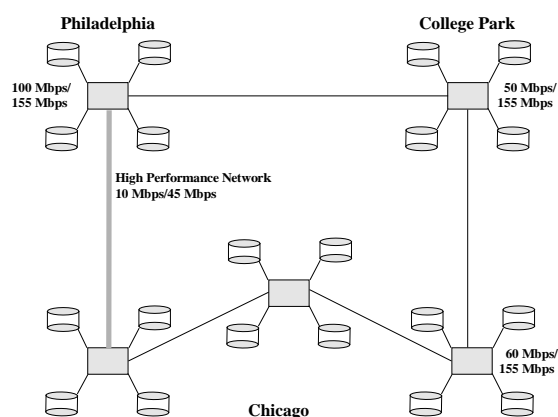


Figure 5. Cluster of workstations in Papyrus.

all into a single system, and more are constantly being developed. Therefore, a DDM system must be able to incorporate new algorithms and methods as needed. For this purpose, BODHI system offers APIs support for creating custom-based agents. Users can easily design and insert their own distributed mining applications with the APIs. The Kensington system adopts *Software Component Architecture*. Software components are software blocks that can be easily combined into a more complex ensemble. In particular, the application server consists of four Enterprise JavaBeans (EJB) classes. Each EJB class provides services to clients that can be accessed through Remote Method Invocation (RMI) calls. High performance software modules in the third-tier server are also components that are integrated via CORBA, RMI or JNI.

The Parallel and Distributed Data Mining Application Suite (PaDDMAS) (Rana, Walker, li, Lynden, & Ward, 2000) is another component based system for developing distributed data mining applications. The overall architecture of PaDDMAS resembles the Kensington system in the sense that it identifies analysis algorithms as object components implemented as either Java or CORBA objects. It also provides a tool set that aids users in creating a distributed data mining application by combining existing data mining components using a dataflow approach. However, PaDDMAS takes one step further to allow easy insertion of custom-based components. To ensure uniformity across components, each component in PaDDMAS has its interface specification written in XML. A user supplied component must also have its interfaces defined in XML. PaDDMAS allows a connection of two components only if their interfaces are compatible. A markup for data mining algorithms has emerged from the Predictive Model Markup Language (PMML) (Grossman et al., 1999). PMML was designed to encode and exchange predictive data mining analysis components like C4.5 (Quinlan, 1993). In that sense, the markup used in PaDDMAS can be considered an attempt to embrace both analysis and data management components. Also, the emphasis of PaDDMAS is on encoding interfaces rather than encoding data structure of components.

Future Directions

Current data mining products are primarily designed for off-line decision support applications. Real-time on-line decision support is a natural extension and it is likely to be one of the primary target applications for the next generation of data mining products. This will require a data mining technology that pays careful attention to the distribution of computing, communication, and storage resources in the environment. Distributed data mining is an ideal candidate for such applications. Several organizations are currently working toward DDM applications in different areas including financial data mining from mobile devices (Kargupta et al., 2002), sensor-network-based distributed database, (Bonnet, Gehrke, & Seshadri, 2001), car-health diagnostics analysis (Wirth, Borth, & Hipp, 2001).

However, DDM still has several major open issues that need to be addressed. First of all, many real-life applications deal with data distribution scenarios that are neither homogeneous nor heterogeneous in their pristine sense described in an earlier section. We may have heterogeneous data sites that share more than one column. We may not have any well defined key that links multiple rows across the sites. We need more algorithms for the heterogeneous scenarios. Also distributed data pre-processing based on metadata needs further explorations.

DDM frequently requires exchange of data mining models among the participating sites. Therefore, seamless and transparent realization of DDM technology will require standardized schemes to represent and exchange models. The Predictive Model Markup Language (PMML), Cross-Industry Standard Process Model for Data Mining (CRISP-DM), other related efforts are likely to be very useful for the development of DDM.

Web search sites like Yahoo and Google are likely to start offering data mining services for analyzing the data they host (Sarawagi & Nagaralu, 2000). Combining the data mining models from such sites will be an interesting DDM application. Since the sites may have partially shared domain, no explicit keys exist linking the data, DDM for this application is also likely to be very challenging.

Acknowledgments

The authors acknowledge supports from the NASA (NRA) NAS2-37143 and the United States National Science Foundation CAREER award IIS-0093353.

References

- Agha, G. (1986). *ACTORS: A model of concurrent computation in distributed systems*. Cambridge, Mass.: MIT Press.
- Agrawal, R., & Shafer, J. C. (1996). Parallel mining of association rules. *IEEE Transactions On Knowledge And Data Engineering*, 8, 962–969.
- Alsabti, K., Ranka, S., & Singh, V. (1997). A one-pass algorithm for accurately estimating quantiles for disk-resident data. In *Proceedings of the VLDB'97 conference*.
- Aronis, J., Kulluri, V., Provost, F., & Buchanan, B. (1997). The WoRLD: Knowledge discovery and multiple distributed databases. In *Proceedingd of florida artificial intellegence research symposium (FLAIRS-97)*.
- Bauer, E., & Kohavi, R. (1999). An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine Learning*, 36(1–2), 105–139.

- Bonnet, P., Gehrke, J., & Seshadri, P. (2001). Towards sensor database systems. In *In proceedings of the second international conference on mobile data management*. Hong Kong.
- Breiman, L. (1999). Pasting small votes for classification in large databases and on-line. *Machine Learning*, 36(1–2), 85–103.
- Byrne, C., & Edwards, P. (1995). Refinement in agent groups. In G. Weiß & S. Sen (Eds.), *Adaption and learning in multi-agent systems* (pp. 22–39). New York: Springer-Verlag. (Proceedings IJCI'95 Workshop, Montreal, Canada, 1995)
- Carmel, D., & Markovitch, S. (1995). Opponent modeling in multi-agent systems. In G. Weiß & S. Sen (Eds.), *Adaption and learning in multi-agent systems* (pp. 40–52). New York: Springer-Verlag. (Proceedings IJCI'95 Workshop, Montreal, Canada, 1995)
- Chan, P., & Stolfo, S. (1993a). Experiments on multistrategy learning by meta-learning. In *Proceeding of the second international conference on information knowledge management* (pp. 314–323).
- Chan, P., & Stolfo, S. (1993b). Toward parallel and distributed learning by meta-learning. In *Working notes aaii work. knowledge discovery in databases* (pp. 227–240). AAAI.
- Chan, P., & Stolfo, S. (1998). Toward scalable learning with non-uniform class and cost distribution: A case study in credit card fraud detection. In *Proceeding of the fourth international conference on knowledge discovery and data mining* (p. o). AAAI Press.
- Chatratchat, J., Darlington, J., Guo, Y., Hedvall, S., Koler, M., & Syed, J. (1999). An architecture for distributed enterprise data mining. In *HPCN europe* (p. 573–582).
- Chen, R., Krishnamoorthy, S., & Kargupta, H. (2001). Distributed web mining using bayesian networks from multiple data streams. In *IEEE international conference on data mining*.
- Chen, R., Krishnamoorthy, S., & Kargupta, H. (2002). Collective mining of bayesian networks from distributed heterogeneous data. *in communication*.
- Cheung, D., Ng, V., Fu, A., & Fu, Y. (1996). Efficient mining of association rules in distributed databases. *IEEE Transaction on Knowledge and Data Engineering*, 8(6), 911–922.
- Cho, V., & Wüthrich, B. (April, 1998). Toward real time discovery from distributed information sources. In *Second pacific-asia conference, PAKKD-98* (pp. 376–377). Melbourne, Australia.
- Davies, R., & Smith, R. (1983). Negotiation as a metaphor for distributed problem solving. *Artificial Intelligence*, 20, 63–108.
- Dhillon, I., & Modha, D. (1999). A data-clustering algorithm on distributed memory multiprocessors. In *Proceedings of the KDD'99 workshop on high performance knowledge discovery*.
- Dietterich, T. G. (2000). An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting and randomization. *Machine Learning*, 40(2), 139–158.
- Durfee, E., Lesser, V., & Corkill, D. (1989). Cooperative distributed problem solving. In A. Barr, P. Cohen, & E. Feigenbaum (Eds.), *The handbook of artificial intelligence, volume iv*. Addison Wesley.
- Fan, W., Stolfo, S., & Zhang, J. (1999). The application of adaboost for distributed, scalable and on-line learning. In *Fifth acm sigkdd international conference on knowledge discovery and data mining*. San Diego, California.
- Finin, T., Labrou, Y., & Mayfield, J. (1997). Kqml as an agent communication language. In J. Bradshaw (Ed.), *Software agents*. MIT Press.

- Forman, G., & Zhang, B. (2000). Distributed data clustering can be efficient and exact. In *Sigkdd explorations* (Vol. 2).
- Freitas, A. A., & Lavington, S. H. (1998). *Mining very large databases with parallel processing*. Kluwer Academic Publishers.
- Gorodetski, V., Skormin, V., Popyack, L., & Karsaev, O. (2000). Distributed learning in a data fusion systems. In *Proceedings the conference of the world computer congress (WCC-2000) intelligent information processing (IIP2000)*. Beijing, China.
- Grossman, R., Bailey, S., Kasif, S., Mon, D., Ramu, A., & Malhi, B. (1998). *The preliminary design of papyrus: A system for high performance, distributed data mining over clusters, meta-clusters and super-clusters*. (Fourth International Conference of Knowledge Discovery and Data Mining, New York, New York, Pages 37–43)
- Grossman, R., Bailey, S., Ramu, A., Malhi, B., Hallstrom, P., Pulleyn, I., & Qin, X. (1999). The management and mining of multiple predictive models using the predictive modeling markup language. *Information and System Technology*, 589–595.
- Grossman, R. L., Bailey, S., Ramu, A., Malhi, B., Sivakumar, H., & Turinsky, A. (1999). Papyrus: A system for data mining over local and wide area clusters and super-clusters. In *Supercomputing IEEE*.
- Grossman, R. L., Bailey, S. M., Sivakumar, H., & Turinsky, A. L. (1999). Papyrus: A system for data mining over local and wide-area clusters and super-clusters. In *Supercomputing IEEE*.
- Guo, Y., & Sutiwaraphun, J. (2000). Distributed learning with knowledge probing: A new framework for distributed data mining. In *Advances in distributed and parallel knowledge discovery*, eds: Hillol Kargupa and Phillip Chan (pp. 115–132). MIT Press.
- Han, E., Karypis, G., & Kumar, V. (1997). Scalable parallel data mining for association rules. In *Proceedings of SIGMOD'97* (pp. 277–288). New York: ACM.
- Hershberger, D., & Kargupta, H. (2001). Distributed multivariate regression using wavelet-based collective data mining. *Journal of Parallel Distributed Computing*, 61, 372–400.
- Hoballah, I., & Varshney, P. (1989). Distributed bayesian signal detection. *IEEE Transactions on information theory*, 35(5), 995–1000.
- Holland, J. H. (1975). *Adaptation in natural artificial systems*. Ann Arbor: University of Michigan Press.
- Honavar, V., Miller, L., & Wong, J. (1998). Distributed knowledge networks. In *Ieee information technology conference*. Syracuse, NY.
- Jensen, V. C., & Soparkar, N. (2000). Frequent itemset counting across multiple tables. In *4th pacific-asia conference on knowledge discovery and data mining*.
- Johnson, E., & Kargupta, H. (1999). Collective, hierarchical clustering from distributed, heterogeneous data. In *Lecture Notes in Computer Science* (Vol. 1759, p. 221–244). Springer-Verlag.
- Joshi, A. (1995). To learn or not to learn. In G. Weiβ & S. Sen (Eds.), *Adaption and learning in multi-agent systems* (pp. 127–139). New York: Springer-Verlag. (Proceedings IJCI'95 Workshop, Montreal, Canada, 1995)
- Joshi, M., Han, E., Karypis, G., & Kumar, V. (2000). Parallel algorithms for data mining. In *Crcp parallel computing handbook*. Morgan Kaufmann.
- Kamath, K., & Musick, R. (2000). Scalable data mining through fine-grained parallelism: The present and the future. In H. Kargupta & P. Chan (Eds.), *Advances in distributed and parallel knowledge discovery*. MIT Press.

- Kargupta, H., Hamzaoglu, I., & Stafford, B. (1997). Scalable, distributed data mining using an agent based architecture. In D. Heckerman, H. Mannila, D. Pregibon, & R. Uthurusamy (Eds.), *Proceedings of knowledge discovery and data mining* (pp. 211–214). Menlo Park, CA: AAAI Press.
- Kargupta, H., Hamzaoglu, I., Stafford, B., Hanagandi, V., & Buescher, K. (1996). PADMA: Parallel data mining agent for scalable text classification. In *Proceedings conference on high performance computing '97* (pp. 290–295). The Society for Computer Simulation International.
- Kargupta, H., Huang, W., Krishnamrthy, S., Park, B., & Wang, S. (2000). Collective principal component analysis from distributed, heterogeneous data. In *Proceedings of the principals of data mining and knowledge discovery*.
- Kargupta, H., Huang, W., S., K., & Johnson, E. (2001). Distributed clustering using collective principal component analysis. *Knowledge and Information Systems Journal Special Issue on Distributed and Parallel Knowledge Discovery*, 3, 422–448.
- Kargupta, H., Park, B., Pittie, S., Liu, L., Kushraj, D., & Sarkar, K. (2002). Mobimine: monitoring the stock market from a pda. *ACM SigKDD Explorations*, 3(2).
- Krishnaswamy, S., Zaslavsky, A., & Loke, S. (2000). An architecture to support distributed data mining services in e-commerce environments. In *Second international workshop on advance issues of e-commerce and web-based information systems (wecwis 2000)*. Milpitas, CA.
- Lam, W., & Segre, A. M. (1997). Distributed data mining of probabilistic knowledge. In *Proceedings of the 17th international conference on distributed computing systems* (pp. 178–185). Washington: IEEE Computer Society Press.
- Lander, S., & Lesser, V. (1992). Customizing distributed search among agents with heterogeneous knowledge. In *Proceedings of the first international conference on information and knowledge management*.
- Lee, W., Stolfo, S., & Mok, K. (1999). A data mining framework for adaptive intrusion detection. In *Proceedings of the 1999 IEEE symposium on security and privacy*.
- Lesser, V., et al.. (1998). Big: A resource bound information gathering agent. In *Proceedings of the fifteenth national conference on artificial intelligence, AAAI'98*.
- Martin, G., Unruh, A., & Urban, S. (1999). *An agent infrastructure for knowledge discovery and event detection* (Tech. Rep. No. MCC-INSL-003-99). Microelectronics and Computer Technology Corporation (MCC).
- McClean, S., Scotney, B., & Greer, K. (2000). Clustering heterogeneous distributed databases. In *Workshop on distributed and parallel knowledge discovery*. Boston, MA, USA.
- McLean, B., Hawkins, C., Spagna, A., Lattanzi, M., Lasker, B., Jenkner, H., & White, R. (1998). New horizons from multi-wavelength sky surveys. *IAU Symposium*. 179.
- Menczer, F., & Belew, R. (1998). Adaptive information agents for distributed textual environments. In K. P. Sycara & M. Wooldridge (Eds.), *Proceedings of the second international conference on autonomous agents* (pp. 157–164). New York: ACM.
- Merz, C. J., & Pazzani, M. J. (1999). A principal components approach to combining regression estimates. *Machine Learning*, 36(1–2), 9–32.
- Minsky, M. (1985). *The society of mind* (1st ed.). Simon and Schuster.
- Mor, Y., Goldman, C., & Rosenschein, J. (1995). Using reciprocity to adapt to others. In G. Weiß & S. Sen (Eds.), *Adaption and learning in multi-agent systems* (pp. 164–176). New York: Springer-Verlag. (Proceedings IJCT'95 Workshop, Montreal, Canada, 1995)

- Moukas, A. (1996). *Amalthea*: Information discovery and filtering using a multiagent evolving ecosystem. In *Proceedings of the conference on practical applications of intelligent agents and multi-agent technology*.
- Newell, A., & Simon, H. (1963). GPS, a program that simulates human thought. *Computers and Thought*, 279–293.
- Nii, P. (1986). Blackboard systems: The blackboard model of problem solving and the evolution of blackboard architectures. *AI Magazine*, 7(2), 38–53.
- Opitz, D., & Maclin, R. (1999). Popular ensemble methods: An empirical study. *Journal of Artificial Intelligence Research*, 11, 169–198.
- Park, B., Ayyagari, R., & Kargupta, H. (2001). A fourier analysis-based approach to learn classifier from distributed heterogeneous data. In *Proceedings of the first siam international conference on data mining*. Chicago, US.
- Park, B., Kargupta, H., Johnson, E., Sanseverino, E., Hershberger, D., & Silvestre, L. (2002). Distributed, collaborative data analysis from heterogeneous sites using a scalable evolutionary technique. *Applied Intelligence*, 16(1).
- Parthasarathy, S. (2001). Towards network-aware data mining. In *4th international workshop on parallel and distributed data mining*. San Francisco, CA, USA.
- Parthasarathy, S., & Ogihara, M. (2000). Clustering distributed homogeneous datasets. In *PDKK* (p. 566-574).
- Parthasarathy, S., & Subramonian, R. (2000). Facilitating data mining on a network of workstations. In K. Kargupta & P. Chan (Eds.), *Advances in distributed and parallel knowledge discovery* (pp. 233–258). AAAI/MIT Press.
- Parthasarathy, S., Zaki, M., Ogihara, M., & Li, W. (2001). Parallel data mining for association rules on shared-memory systems. *Knowledge and Information Systems*, 3(1), 1-29.
- Pokrajac, D., Fiez, T., Obradovic, D., Kwek, S., & Obradovic, Z. (1999). Distribution comparison for site-specific regression modeling in agriculture. In *Proceedings of the international joint conference on neural networks*.
- Prodromidis, A., & Stolfo, S. (2000). Cost complexity-based pruning of ensemble classifiers. In *Workshop on distributed and parallel knowledge discovery at KDD-2000* (pp. 30–40). Boston.
- Provost, F. J., & Buchanan, B. (1995). Inductive policy: The pragmatics of bias selection. *Machine Learning*, 20, 35–61.
- Quinlan, J. R. (1993). *C4.5: Programs for machine learning*. Morgan Kaufman.
- Rana, O., Walker, D., li, M., Lynden, S., & Ward, M. (2000). Paddmas: Parallel and distributed data mining application suite. In *Fourteenth international parallel and distributed processing symposium* (pp. 387–392). Cancun, Mexico.
- Rosenschein, J. (1994). Designing conventions for automated negotiation. *AI Magazine*, 29-46.
- Rumelhart, D., & McClelland, J. (1986). *Parallel distributed processing: Explorations in the microstructure of cognition; vol. 1: Foundations: Vol. 2: Psychological and biological models*. Cambridge, Mass.: MIT Press.
- Samatova, N., Ostrouchov, G., Geist, A., & Melechko, A. (2002). Ratchet: An efficient cover-based merging of clustering hierarchies from distributed datasets. *An International Journal of Distributed and Parallel Databases*, 11(2), 157–180.

- Sandholm, T., & Crites, R. (1995). On multiagent q-learning in a semi-competitive domain. In G. Weiß & S. Sen (Eds.), *Adaption and learning in multi-agent systems* (pp. 191–205). New York: Springer-Verlag. (Proceedings IJCI'95 Workshop, Montreal, Canada, 1995)
- Sarawagi, S., & Nagaralu, S. (2000). Data mining models as services on the internet. *SIGKDD Explorations*, 2(1), 24–28.
- Sayal, M., & Scheuermann, P. (2000). A distributed clustering algorithm for web-based access patterns. In *Workshop on distributed and parallel knowledge discovery at KDD-2000* (pp. 41–48). Boston.
- Schuster, A., & Wolff, R. (2001). Communication efficient distributed mining of association rules. In *Acm sigmod*. Santa Barbara.
- Sen, S. (1997). Developing an automated distributed meeting scheduler. *IEEE Expert*, 12(4), 41–45.
- Sen, S., & Sekaran, M. (1995). Multiagent coordination with learning classifier systems. In G. Weiß & S. Sen (Eds.), *Adaption and learning in multi-agent systems* (p. 218–233). New York: Springer-Verlag. (Proceedings IJCI'95 Workshop, Montreal, Canada, 1995)
- Smith, R. (1980). The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Transactions on Computers*, C-12(12), 1104–1113.
- Stolfo, S., et al.. (1997). Jam: Java agents for meta-learning over distributed databases. In *Proceedings third international conference on knowledge discovery and data mining* (pp. 74–81). Menlo Park, CA: AAAI Press.
- Szalay, A. (1998). The evolving universe. *ASSL*(231).
- Ting, K., & Low, B. (1997). Model combination in the multiple-data-base scenario. In *9th european conference on machine learning* (pp. 250–265).
- Tumer, K., & Ghosh, J. (2000). Robust order statistics based ensemble for distributed data mining. In *Advances in distributed and parallel knowledge discovery, eds: Kargupta, hillol and chan, philip* (pp. 185–210). MIT.
- Turinsky, A. L., & Grossman, R. L. (2000). A framework for finding distributed data mining strategies that are intermediate between centralized strategies and in-place strategies. In *Workshop on distributed and parallel knowledge discovery*. Boston, MA, USA.
- Viswanathan, R., & Varshney, P. (1997). Distributed detection with multiple sensors. *Proceedings of IEEE*, 85, 54–63.
- Weiß, G. (1995). Adpation and learning in multi-agent systems: Some remarks and a bibliography. In G. Weiß & S. Sen (Eds.), *Adaption and learning in multi-agent systems* (pp. 1–21). New York: Springer-Verlag. (Proceedings IJCI'95 Workshop, Montreal, Canada, 1995)
- Wirth, R., Borth, M., & Hipp, J. (2001). When distribution is part of the semantics: A new problem class for distributed knowledge discovery. In *Proceedings of PKDD-2001 workshop on ubiquitous data mining for mobile and distributed environments*. Freiburg, Germany.
- Wolfe, M. (1995). *High performance compilers for parallel computing*. Addison Wesley.
- Wolpert, D. (1992). Stacked generalization. *Neural Networks*, 5, 241–259.
- Wooldridge, M., & Jennings, N. R. (1995). Intelligent agent: Theory and practice. *Knowledge Engineering Review*, 10(2).
- Wooldridge, M., & Jenneings, N. (1995). *Intelligent agents: ECAI-94 workshop on agent theories, architectures, and languages*. New York, NY: Springer-Verlag.

- Wüthrich, B., Cho, V., Pun, J., & Zhang, J. (2000). Data quality in distributed environments. In *Advances in distributed and parallel knowledge discovery*, eds: *Hillol Kargupa and Phillip Chan* (pp. 295–316). MIT Press.
- Yamanishi, K. (1997). Distributed cooperative bayesian learning strategies. In *Proceedings of colt 97* (pp. 250–262). New York: ACM.
- Zaki, M. J. e. a. (1996). Parallel data mining for association rules on shared memory multi-processors. In *Supercomputng '96*.
- Zaki, M. J. e. a. (1997). A localized algorihm for parallel association mining. In *9th acm symp. parallel algorithms and architectures* (p. Not available).
- Zhang, B., Hsu, M., & Forman, G. (2000). Accurate recasting of parameter estimation algorithms using sufficient statistics for efficient parallel speed-up: Demonstrated for center-based data clustering algorithms. In *PKDD*.