Distributed Data Mining: An Overview

Yongjian Fu
Department of Computer Science
University of Missouri-Rolla
yongjian@umr.edu

Abstract

This article gives a basic introduction to distributed data mining. We explain what distribution data mining is and why distributed data mining is interesting. Problems and progress in distributed data mining are also discussed.

1 Introduction

Facing a rapidly increasing amount of data, businesses and organizations are turning to data mining to make the most out of their data. Data mining is the process of identifying interesting patterns in large data sets. There has been a lot of progress and success in both data mining research and data mining applications.

Recently, distributed data mining has attracted a lot of attention among data mining community [5, 7, 9, 13, 14, 15]. Distributed data mining refers to the mining of distributed data sets. The data sets are stored in local databases, hosted by local computers, which are connected through a computer network. Data mining takes place at a local level and at a global level where local data mining results are combined to gain global findings. A simple architecture for distributed data mining is shown in Fig. 1.

Distributed data mining is often mentioned with parallel data mining in literature. While both attempt to improve the performance of traditional data mining systems, they assume different system architectures and take different approaches. In distributed data mining, computers are distributed and communicate through message passing. In parallel data mining, a parallel computer is assumed with processors sharing memory and/or disk. Computers in a distributed data mining system may be viewed as processors sharing nothing. This difference in architecture greatly influences algorithm design, cost model, and performance measure in distributed and parallel data mining. In this article, we focus on distributed data mining.

2 Why Distributed Data Mining?

Until recently, data mining research has been focusing on centralized data mining where a centralized data set is stored at a single site. Although a lot of progress has been made toward mining centralized data sets, the techniques are inefficient or incapable to deal with distributed data sets. The following reasons have made distributed data mining interesting.

- Distributed data.

  In some applications, data are inherently distributed, but it is necessary to gain global insights from the distributed data sets. For example, each site of a multinational company manages its own operational data locally, but the data must be analyzed for global patterns to allow company-wide activities such as planning, marketing, and sales.

  The straightforward solution is to transfer all data to a central site, where data mining is done. However, even if we can find a central site with enough capacity to handle the data storage and data mining, it may be too long or too expensive to transfer the local data sets because of their sizes. Sometimes, the local data sets cannot be transferred because of security or autonomy of the data sets.

- Performance and scalability.

  Besides dealing with distributed data, distributed data mining may be useful where data is stored at a single site. One scenario is that the data set is so large that it is beyond the data mining capability of the site. In such a case, the site may send part of the data set to other sites. The involved sites perform data mining and the results are then combined.

  Even if the site is capable of mining the data set, it may be worthwhile to send part or all of the

data set to other sites. Each site may take a slide of the data set and mine it (data distribution), or each site may perform a part of mining task (task distribution) on a replicated data set. Either way, the results from all sites are combined. This will be much faster than centralized data mining since the work load is distributed among the sites.

The above discussion has demonstrated that distributed data mining techniques are scalable. A data mining technique is scalable when its performance does not degrade much with the increase of data set size. In a distributed data mining system, the number of sites involved in a task can grow or shrink with the data set size. The overall performance of such a system remains steady.

Not only distributed data mining is a desirable approach in many cases, it is feasible with the advances in computer network, especially the growth of Internet and intranets.

Of course, distributed data mining does not come with no cost. Compared with centralized data mining, the techniques in distributed data mining are more complex. Careful design is required for a distributed data mining task.

3 Issues in Distributed Data Mining

Simply treating local data sets as a part of a centralized global database and applying centralized data mining techniques on such a database works poorly for distributed data mining because of the distributed nature of data. One has to look at the characteristics of data, computer network, and data mining task to design an appropriate solution for a distributed data mining task.

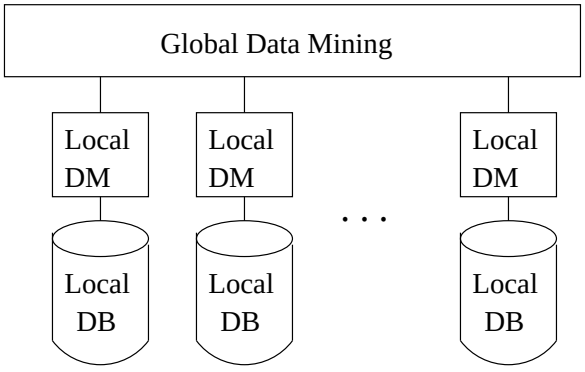

Figure 1: Distributed Data Mining Architecture.

Some of the issues that should be considered in distributed data mining are listed below.

- Homogeneous vs. heterogeneous data.

  Most studies on distributed data mining assume that local databases are homogeneous, that is, local databases are on the same platform, managed by the same DBMS with the same schema, and the corresponding attributes have the same domain. If the local databases are heterogeneous, data mining systems must be adapted to work on local databases. Moreover, the local schemas should be integrated into a global schema first. Otherwise, it is hard if not impossible to interpret the results. Furthermore, any conflicts among attributes should be resolved.

- Fragmentation of data.

  A local table in a local database may be or can be viewed as a fragment of a global table. Fragmentation may be vertical or horizontal. In a horizontal fragmentation, each fragment is a subset of tuples in a global table and the fragments share the same schema. In a vertical fragmentation, each fragment has a subset of attributes of a global table. A mixture of the two is also possible where some tables are vertically fragmented and others are horizontally fragmented. Fragmentation may also be nested where a fragment is further fragmented.

- Replication of data.

  Some or all data in a local database may be replicated at other sites. Replication improves availability of data, but also makes it harder to maintain data consistency. Normally, data replication is not done for the purpose of data mining, rather it is a decision based on business or computing needs. It is, however, possible to replicate data for the purpose of data mining. In that case, the data miner has to decide what data or part of data to replicate.

- Communication cost.

  In centralized data mining, the main concern for the efficiency of a data mining algorithm is its I/O and/or CPU time. In a distributed environment, one has to consider the communication cost. For a slow network, the communication cost will dominate the overall cost. The communication cost is determined by the network bandwidth and the number of messages that are

sent across the network. A cost model different from that in centralized data mining is needed for distributed data mining.

- Integration of results.

  The integration is not simple putting together results from all sites. An interesting pattern in a local database may not be interesting globally. For example, a frequent itemset at a local site may be infrequent globally. Since the goal of distributed data mining is to find globally interesting patterns, the patterns and their properties (interestingness) should be collected from all sites and verified globally for their interestingness.

- Data skewness.

  The statistical distributions of data, such as attribute values and class memberships, are usually different among local databases. A local model obtained by mining a local database is unavoidably affected by such distribution. Such data skewness can make the local models inaccurate, sometimes even useless. For example, a classifier learned from a local database that has no or few instances of a class will not be able to classify future instance of that class.

These issues are not isolated, rather they are interrelated. For examples, a vertical fragmentation of a global table will cause the local databases to be heterogeneous. Horizontal data fragmentation may cause data skewness if not done carefully. Replication of fragments or tables may reduce communication cost for data access, however, may increase communication cost in data consistency maintenance.

There are many other factors that need to be considered in distributed data mining, such as security, privacy, and autonomy of local databases, network topology and transmission scheme, and work load of each site.

4 Progress in Distributed Data Mining

Distributed data mining has attracted growing interests from researchers recently. A number of distributed algorithms have been developed for classification, association, and clustering. Some general frameworks for distributed data mining have also be proposed.

4.1 Distributed Classification

Classification is the derivation of a function or model, called a classifier, which determines the class of an object based on its attributes. A set of objects is given as the training set in which every object is represented by a vector of attributes along with its class. A classifier is constructed by analyzing the relationship between the attributes and the classes of the objects in the training set. Such a classifier can be used to predict the classes of future objects and to develop a better understanding of the classes of the objects in the database. For example, from a set of diagnosed patients, who serve as the training set, a classifier can be built, which decides a patient's disease from his/her diagnostic data, such as age, sex, weight, temperature, blood pressure, etc.

Prodromidis et. al. have developed an architecture and a method for building meta-classifiers using Java agents [12]. A meta-classifier is a classifier that is learned from base classifiers, which are induced from local databases. Different techniques can be used to learn the base classifiers. After the base classifiers are learned, they are treated as black boxes when they are combined by the meta-classifier. A validation data set is classified by the base classifiers, whose predictions along with the data set are used to train the meta-classifier. For each object in the validation data set, the meta-classifier classifies it by combining the predictions from base classifiers. Different combining strategies are available, including voting, arbitrating, and Bayesian summarization. They also designed a distributed architecture in which each site is autonomous. Sites cooperate with each other by sharing base classifiers as well as meta-classifiers, which are implemented as Java agents.

Guo and Sutiwaraphun have proposed another distributed approach for classification [6]. Like Prodromidis et. al., their approach uses two-level learning, local learning and meta learning which combines local learners. An local learner is similar to a base classifier in [12] and the meta learner is similar to the meta-classifier in [12]. However, their goal is to find a global distribution of data from local distributions. Basically, the parameters in the global distribution are to be computed from local distributions. To deal with data skewness, Guo and Sutiwaraphun have developed a redistribution method which replicates data that are contradictorily classified among the local learners.

4.2 Distributed Association

Association rules were first introduced in [1] for the discovery of associations among items in a transac-

tional database. For example, in a telecommunication database, an association rule that "call waiting" is associated with "call display", denoted as "call waiting → call display", says if a customer subscribes to the "call waiting" service, he or she very likely also has "call display". To find interesting association rules, two measures, *support* and *confidence*, have been introduced. The *support* of a rule is the frequency of the set (called *itemset*) which consists of all items in the rule, i.e., the probability a transaction contains the itemset. The confidence of a rule is the probability that a transaction contains the items in the right hand side of the rule when the transaction contains the left hand side items of the rule. The task is to find all association rules whose support and confidence are above the given support threshold, *min_sup*, and the given confidence threshold *min_conf*, respectively.

The most popular algorithm for mining association rules is the *Apriori* algorithm [2]. Apriori finds all association rules in two steps. First, frequent itemsets, i.e., itemsets whose support is no less than min_sup, are identified iteratively by examining itemsets of site k at the k-th iteration. Second, association rules are derived from the frequent itemsets and filtered out using min_conf. The emphasis has been on the first step, which dominates computationally. Apriori exploits the basic property of a frequent itemset: all subsets of a frequent itemset must be frequent. Starting with singleton itemsets, Apriori computes their supports by scanning the database, and filters out frequent itemsets. At the end of each iteration, only itemsets whose immediate subsets are all frequent at the current iteration are considered at the next iteration.

Cheung et. al. have developed a distributed algorithm for mining association rules [4]. Like most association mining algorithms, it focuses on the generation of (globally) frequent itemsets. The most significant feature of the algorithm is that it exploits the following important property: a globally frequent itemset must be frequent in one of the local databases. At each local site, local frequent itemsets are computed by the Apriori algorithm [2]. At each iteration, a polling site is decided for every itemset which is frequent at a site. Each site will send supports of itemsets to their polling sites which will compute their global supports and broadcast globally frequent itemsets to other sites. The number of messages required is reduced since only itemsets which are frequent at a site need to be considered for next iteration and only one site is polling the support for each of these itemsets.

4.3 Distributed Clustering

Clustering is the identification of classes, also called clusters or groups, for a set of objects whose classes are unknown. The objects are so clustered that the intraclass similarities are maximized and the interclass similarities are minimized, based on some criteria defined on the attributes of the objects. Once the clusters are decided, the objects are labeled with their corresponding clusters, and common features of the objects in a cluster are summarized to form the class description. Clustering has been studied extensively in statistics, machine learning, pattern recognition, and database systems, with a large number of clustering algorithms developed.

A family of agglomerative hierarchical clustering methods have been developed in statistics. Starting with single object clusters, an agglomerative hierarchical clustering method consecutively merges the two closest clusters until there is only one cluster left. This results in a hierarchical clustering, called dendrogram.

A distributed clustering algorithm has been proposed by Johnson and Kargupta [8]. They assume the data is vertically fragmented, i.e., each site stores a subset of the attributes of objects. Furthermore, they assume a unique key of objects is stored at every site. At a local site, an agglomerative hierarchical clustering algorithm is employed to find the local dendrogram. The local dendrograms are then transferred to a synthesisor, called facilitator. At facilitator, the objects are clustered based on an estimated distance that is calculated from local dendrograms.

4.4 Distributed Architecture

Some studies focus on architectures for distributed data mining. An architecture lays out the components in a distributed data mining system and their interrelationships.

Kargupta et. al. have proposed a framework for mining of distributed, fragmented data [10]. They propose to used a set of orthonormal representations for the sought data model. Each local site mines the local data to get a local representation that uses the orthonormal representations as the basis. The local representations are then integrated into the global model. An earlier study has shown that Wavelet and Fourier functions are good candidates for the orthonormal representations.

Chattratichat et. al. have developed an Enterprise Java Bean based architecture for distributed data mining [3]. Their Kensington data mining system uses several Enterprise Java Beans to provide services such as data mining, object management, and

storage management for a possibly remote client. The client is implemented as Java Beans and can connect to server through the RMI protocol.

Krishnaswamy et. al. have proposed to build distributed data mining management system for e-commerce [11]. The main modules in their system include user manager, algorithm manager, mining process manager, and agent control center that manages various agents for data mining, networking, and data resource monitoring and so on.

5 Conclusions

Distributed data mining is the mining of distributed data. It intends to obtain global knowledge from local data at distributed sites. Distributed data mining has attracted increasing interests because of its potentials in dealing with distributed data and its performance advantages. Much progress has been seen in distributed classification, association, clustering, and architecture. For the increasing amount of distributed data, distributed data mining provides a promising solution. With the development of high speed computer networks, distributed data mining is a viable choice for high performance data mining.

Although there have been lots of success in distributed data mining, it is still a young area. There are no satisfactory solutions for many problems, such as schema integration, data skewness, interaction and interdependency among sites, and integration of results. There are also problems, which are common in centralized data mining, such as privacy, interestingness measure, and user interface. More research and development is needed to fully understand and to take advantage of distributed data mining.

References

[1] R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In *Proc. 1993 ACM-SIGMOD Int. Conf. Management of Data*, pages 207–216, Washington, D.C., May 1993.

[2] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *Proc. 1994 Int. Conf. Very Large Data Bases*, pages 487–499, Santiago, Chile, September 1994.

[3] J. Chattratichat, J. Darlintong, Y. Guo, S. Hedvall, M. Kohler, and J. Syed. An architecture for distributed enterprise data mining. In *Proc. 7th Int'l Conf. on High-Performance Computing and Networking*, April 1999.

[4] D. Cheung, V. Ng, A. Fu, and Y. Fu. Efficient mining of association rules in distributed databases. *IEEE Trans. on Knowledge and Data Engineering*, 8:911–922, 1996.

[5] Y. Guo and R. Grossman. Scalable parallel and distributed data mining. *Data Mining and Knowledge Discovery*, 3, Sept. 1999.

[6] Y. Guo and J. Sutiwaraphun. Distributed classification with knowledge probing. In H. Kargupta and P. Chan, editors, *Advances in Distributed and Parallel Knowledge Discovery*. AAAI Press, 2000.

[7] IEEE IPDPS. *Workshop on High Performance Data Mining*. http://www.cs.rpi.edu/ zaki/HPDM/, 2000.

[8] E. Johnson and H. Kargupta. Collective, hierarchical clustering from distributed, heterogeneous data. In M. Zaki and C. Ho, editors, *Large-Scale Parallel KDD Systems, Volumn 1759*. Springer-Verlag, 2000.

[9] H. Kargupta and P. Chan. *Advances in Distributed and Parallel Knowledge Discovery*. AAAI Press, 2000.

[10] H. Kargupta, B. Park, D. Hershberger, and E. Johnson. Collective data mining: A new perspective toward distributed data mining. In H. Kargupta and P. Chan, editors, *Advances in Distributed and Parallel Knowledge Discovery*. AAAI Press, 2000.

[11] S. Krishnaswamy, A. Zaslavsky, and S. Loke. An architecture to support distributed enterprise data mining services in e-commerce environments. In *Workshop on Advanced Issues of E-Commerce and Web Based Information Systems*, pages 239–246, 2000.

[12] A. Prodromidis, P. Chan, and S. Stolfo. Meta-learning in distributed data mining systems: Issues and approaches. In H. Kargupta and P. Chan, editors, *Advances in Distributed and Parallel Knowledge Discovery*. AAAI Press, 2000.

[13] ACM SIGKDD. *Workshop on Distributed Data Mining*. http://www.eecs.wsu.edu/~hillol/DDMWS/papers.html, 1998.

[14] ACM SIGKDD. *Workshop on Distributed and Parallel Data Mining*. http://www.eecs.wsu.edu/~hillol/DKD/dpks2000.html, 2000.

[15] M. Zaki. Parallel and distributed association mining: A survey. *IEEE Concurrency*, 7:14–25, 1999.