IMPACT: International Journal of Research in Engineering & Technology (IMPACT: IJRET)
ISSN(E): 2321-8843; ISSN(P): 2347-4599
Vol. 2, Issue 3, Mar 2014, 107-120

A SURVEY ON DISTRIBUTED DATA MINING AND ITS TRENDS

S. V. S. GANGA DEVI

Professor in MCA, K.S.R.M. College of Engineering, Kadapa, Andhra Pradesh, India

ABSTRACT

Data mining technology has emerged as a means for identifying patterns and trends from large quantities of data. The Data Mining technology normally adopts data integration method to generate Data warehouse, on which to gather all data into a central site, and then run an algorithm against that data to extract the useful Module Prediction and knowledge evaluation. However, a single data-mining technique has not been proven appropriate for every domain and data set. Data mining techniques involving in such complex environment must encounter great dynamics due to changes in the system can affect the overall performance of the system. Distributed data mining is originated from the need of mining over decentralized data sources. The field of Distributed Data Mining (DDM) deals with these challenges in analyzing distributed data and offers many algorithmic solutions to perform different data analysis and mining operations in a fundamentally distributed manner that pays careful attention to the resource constraints. This paper is a survey concerned with Distributed Data Mining algorithms, methods and trends in order to discover knowledge from distributed data in an effective and efficient way.

KEYWORDS: Distributed Data Mining, Grid Computing, Ensemble Learning, Multi Agent Systems

INTRODUCTION

The continuous developments in information and communication technology have recently led to the appearance of distributed computing environments, which comprise several, and different sources of large volumes of data and several computing units. The most prominent example of a distributed environment is the Internet, where increasingly more databases and data streams appear that deal with several areas, such as meteorology, oceanography, economy and others. In addition the Internet constitutes the communication medium for geographically distributed information systems, as for example the earth observing system of NASA (eos.gsfc.nasa.gov). Other examples of distributed environments that have been developed in the last few years are sensor networks for process monitoring and grids where a large number of computing and storage units are interconnected over a high-speed network.

The application of the classical knowledge discovery process in distributed environments requires the collection of distributed data in a data warehouse for central processing. However, this is usually either ineffective or infeasible for the following reasons:

- **Storage Cost:** It is obvious that the requirements of a central storage system are enormous. A classical example concerns data from the astronomy science, and especially images from earth and space telescopes. The size of such databases is reaching the scales of exabytes (10^{18} bytes) and is increasing at a high pace. The central storage of the data of all telescopes of the planet would require a huge data warehouse of enormous cost.
- **Communication Cost:** The transfer of huge data volumes over network might take extremely much time and also require an unbearable financial cost. Even a small volume of data might create problems in wireless network

environments with limited bandwidth. Note also that communication may be a continuous overhead, as distributed databases are not always constant and unchangeable. On the contrary, it is common to have databases that are frequently updated with new data or data streams that constantly record information (e.g. remote sensing sports statistics, etc.).

- **Computational Cost:** The computational cost of mining a central data warehouse is much bigger than the sum of the cost of analyzing smaller parts of the data that could also be done in parallel. In a grid, for example, it is easier to gather the data at a central location. However, a distributed mining approach would make a better exploitation of the available resources.
- **Private and Sensitive Data:** There are many popular data mining applications that deal with sensitive data, such as people's medical and financial records. The central collection of such data is not desirable as it puts their privacy into risk. In certain cases (e.g. banking, telecommunication) the data might belong to different, perhaps competing, organizations that want to exchange knowledge without the exchange of raw private data.

Distributed Data Mining (DDM) (Fu, 2001; Park & Kargupta, 2003) is concerned with the application of the classical Data Mining procedure in a distributed computing environment trying to make the best of the available resources (communication network, computing units and databases). Data Mining takes place both locally at each distributed site and at a global level where the local knowledge is fused in order to discover global knowledge. A typical architecture of a DDM approach is depicted in Figure 1. The first phase normally involves the analysis of the local database at each distributed site. Then, the discovered knowledge is usually transmitted to a merger site, where the integration of the distributed local models is performed. The results are transmitted back to the distributed databases, so that all sites become updated with the global knowledge. In some approaches, instead of a merger site, the local models are broadcasted to all other sites, so that each site can in parallel compute the global model. Distributed databases may have homogeneous or heterogeneous schemata. In the former case, the attributes describing the data are the same in each distributed database. This is often the case when the databases belong to the same organization (e.g. local stores of a chain). In the latter case the attributes differ among the distributed databases. In certain applications a key attribute might be present in the heterogeneous databases, which will allow the association between tuples. In other applications the target attribute for prediction might be common across all distributed databases.

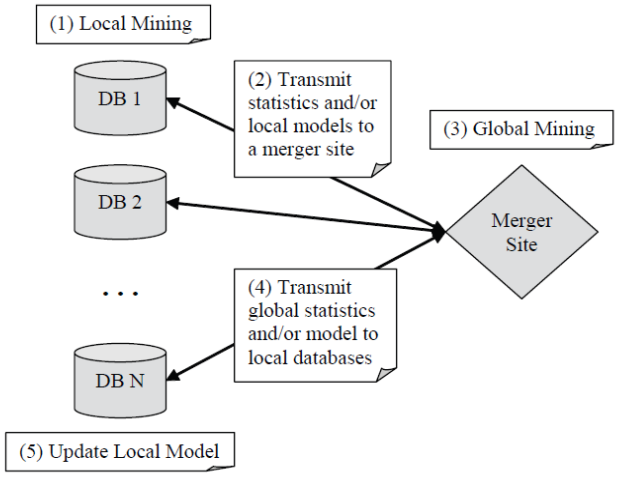

Figure 1: Typical Architecture of Distributed Data Mining Approaches

DDM ON GRID

The Grid is a distributed computing infrastructure that enables coordinated resource sharing within dynamic organizations consisting of individuals, institutions, and resources. The main aim of grid computing is to give organizations and application developers the ability to create distributed computing environments that can utilize computing resources on demand. Grid computing can leverage the computing power of a large numbers of server computers, desktop PCs, clusters and other kind of hardware. Therefore, it can help increase efficiencies and reduce the cost of computing networks by decreasing data processing time and optimizing resources and distributing workloads, thereby allowing users to achieve much faster results on large operations and at lower costs.

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

DDM TECHNIQUES

The increasing demand to scale up to massive data sets inherently distributed over a network with limited bandwidth and computational resources available motivated the development of the techniques of DDM. A number of approaches and techniques have been proposed in literatures.

Some data mining techniques can be used to adapt DDM. Bayesian methods were developed in the framework of statistics for many years. Last ten years, they were applied in the problems of data mining. Decision tree is well-known in data mining. Decision tree technique has been used in DDM. Some statistical techniques such as bagging, boosting and stacking etc., would be extended to combine local models in a distributed environment. The techniques such as Multi-agent Systems, ensemble learning, similarity-based and collective data mining [10] are presented in DDM literatures. This section mainly present the DDM techniques based on Multi-agent Systems and ensemble learning.

Agent-Based

Multi-Agent Systems (MAS) is a system composed of several agents, capable of reaching goals that are difficult to achieve by an individual system. MAS is the emerging sub field of artificial intelligence that aims to provide both principles for construction of complex systems involving multiple agents and mechanisms for coordination of independent agents' behaviors. Several efforts have been devoted to enable DDM through Mass. In [3] the authors present a MAS for context-based distributed data mining. MAS is fundamentally designed for collaborative problem solving in distributed environments. An agent-based data mining system is a natural choice for mining large sets of inherently distributed data. Many DDM system such as JAM [4], are based on multi-agent techniques.

The authors describe a parallel/distributed data mining system PADMA (parallel Data Mining Agents) that uses software agents for local data accessing and analysis and a Web based interface for interactive data visualization.

PADMA has been used in medical applications. An agent-based meta-learning system for large-scale data mining applications, which is called JAM (Java Agents for Meta-learning), is described. JAM was empirically evaluated against real credit card transaction data where the target data mining application was to compute predictive models that detect fraudulent transactions. However, these works are focusing on one of the many steps in data mining. Papyrus is a Java-based system addressing wide-area distributed data mining over clusters of heterogeneous data sites and meta-clusters. It supports different task and predictive model strategies including C4.5. Mobile data mining agents move data, intermediate results, and models between clusters to perform all computation locally and reduce network load, or from local sites to a central root which produces the final result. Each cluster has one distinguished node which acts as its cluster access and control point for the agents. Coordination of the overall clustering task is either done by a central root site or distributed to the (peer-to-peer) network of cluster access points. Papyrus supports various methods for combining and exchanging the locally mined predictive models and metadata required to describe them by using a special markup language. Klusch et al. also proposed a kernel density estimation based clustering scheme for agent-based distributed data clustering [1].

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

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

Ensemble Learning

Ensemble methods are gaining more and more attention in the machine-learning and data mining communities. By definition, an ensemble is a group of learning models whose predictions are aggregated to give the final prediction. It is widely accepted that an ensemble is usually better than a single classifier given the same amount of training information. A number of effective ensemble generation algorithms have been invented during the past decade, such as bagging (Breiman, 1996), boosting Freund and Schapire, 1996), arcing (Breiman, 1998) and random forest (Breiman, 2001). The effectiveness of the ensemble methods relies on creating a collection of diverse, yet accurate learning models. Two costs associated with ensemble methods are that they require much more memory to store all the learning models, and it takes much more computation time to get a prediction for an unlabeled data point. Although these extra

costs may seem to be negligible with a small research data set, they may become serious when the ensemble method is applied to a large scale real-world data set. In fact, a large scale implementation of ensemble learning can easily generate an ensemble with thousands of learning models (Street and Kim, 2001).

A number of effective ensemble generation algorithms have been invented during the past decade, such as bagging (Breiman, 1996), boosting (Freund and Schapire, 1996), arcing (Breiman, 1998) and random forest (Breiman, 2001). The effectiveness of the ensemble methods relies on creating a collection of diverse, yet accurate learning models. Ensemble-based distributed data-mining techniques enable large companies (like Wal Mart) that store data at hundreds of different locations to build learning models locally and then combine all the models for future prediction and knowledge discovery.

The storage and computation time will become non-trivial under such circumstances. There are two main advantages of DDM using ensembles. The first advantage can be obviously seen when the local model is much smaller than the local data: sending only the model thus reduces the load on the network and the network bandwidth requirement. The second one is that sharing only the model, instead of the data, gains reasonable security for some organizations since it overcomes issues of privacy. 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 (weighted or un weighted) schemes are employed to aggregate base classifiers.

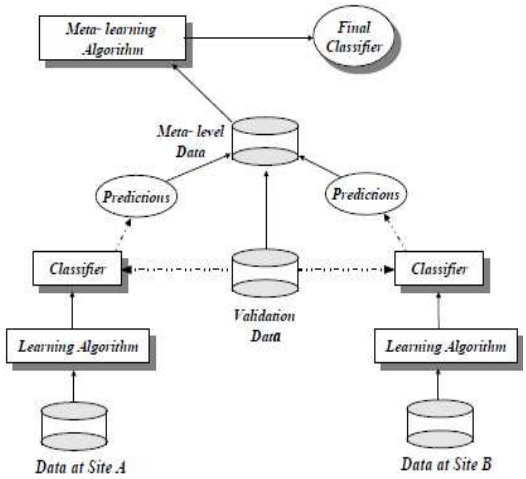

Figure 2: Meta Learning from Distributed Homogeneous Data Sites

The ensemble aproach 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 aplied 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

- Concrete base classifiers at each site using a classifier learning algorithms.
- 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.
- 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.

- **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. It there is a tie, the arbiter's prediction gets the preference.
- **The Combiner Scheme:** The combiner schme 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 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 & Sutiwaaphun, 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:

- Generate base classifiers at each site using off-the-shelf classifier learning algorithms.
- Select a set of unlabeled data for the probing set.
- Prepare probing data set by combining predictions from all base classifiers.
- 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 classify.

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 re3ason, it is generally believed that mining of heterogeneous distributed data is more challenging. The issues in mining from heterogeneous data is discussed in (Provost & Buchaman, 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 requries 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 variabnce models generated from heterogeneous sites. The techniques works by ordering the predictions of different classifiers and using them in an appropriate manner.

The paper gives seveal 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 analyze 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 classifeirs. 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 constructured from it. When a combination of local classifier can not classify an unseen data with a high confidecne, 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).

Distributed Association Rule Mining

Agrawal and Shafer (1996) discuss three parallel algorithms for mining association rules. One of those, the Count Distribution (CD) algorithm, focuses on minimizing the communication cost, and is therefore suitable for mining association rules in a distributed computing environment. CD uses the Apriori algorithm (Agrawal and Srikant, 1994) locally at each data site. In each pass *k* of the algorithm, each site generates the same candidate *k*-itemsets based on the globally frequent itemsets of the previous phase. Then, each site calculates the local support counts of the candidate itemsets and broadcasts them to the rest of the sites, so that global support counts can be computed at each site. Subsequently, each site computes the *k*-frequent itemsets based on the global counts of the candidate itemsets. The communication complexity of CD in pass *k* is *O*(|*Ck*|*n2*), where *Ck* is the set of candidate *k*-itemsets and *n* is the number of sites. In addition, CD involves a synchronization step when each site waits to receive the local support counts from every other site.

Another algorithm that is based on Apriori is the Distributed Mining of Association rules (DMA) algorithm (Cheung, Ng, Fu & Fu, 1996), which is also found as Fast Distributed Mining of association rules (FDM) algorithm in (Cheung, Han, Ng, Fu & Fu, 1996). DMAgenerates a smaller number of candidate itemsets than CD, by pruning at each site the itemsets that are not locally frequent. In addition, it uses polling sites to optimize the exchange of support counts among sites, reducing the communication complexity in pass k to $O(|C_k|n)$, where C_k is the set of candidate k-itemsets and *n* is the number of sites. However, the performance enhancements of DMA over CD are based on the assumption that the data distributions at the different sites are skewed. When this assumption is violated, DMA actually introduces a larger overhead than CD due to its higher complexity.

The Optimized Distributed Association rule Mining (ODAM) algorithm (Ashrafi, Taniar & Smith, 2004) follows the paradigm of CD and DMA, but attempts to minimize communication and synchronization costs in two ways. At the local mining level, it proposes a technical extension to the Apriori algorithm. It reduces the size of transactions by: i) deleting the items that weren't found frequent in the previous step and ii) deleting duplicate transactions, but keeping track of them through a counter.

It then attempts to fit the remaining transaction into main memory in order to avoid disk access costs. At the communication level, it minimizes the total message exchange by sending support counts of candidate itemsets to a single site, called receiver. The receiver broadcasts the globally frequent itemsets back to the distributed sites.

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, Ostrochov, 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 iterative 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 Kullback0Leiber 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.

Database Clustering

Real-world, physically distributed databases have an intrinsic data skewness property. The data distributions at different sites are not identical. For example, data related to a disease from hospitals around the world might have varying distributions due to different nutrition habits, climate and quality of life. The same is true for buying patterns identified in supermarkets at different regions of a country. Web document classifiers trained from directories of different Web portals is another example.

Neglecting the above phenomenon, may introduce problems in the resulting knowledge. If all databases are considered as a single logical entity then the idiosyncrasies of different sites will not be detected. On the other hand if each database is mined separately, then knowledge that concerns more than one database might be lost. The solution that several researchers have followed is to cluster the databases themselves, identify groups of similar databases, and apply DDM methods on each group of databases.

Parthasarathy and Ogihara (2000) present an approach on clustering distributed databases, based on association rules. The clustering method used, is an extension of hierarchical agglomerative clustering that uses a measure of similarity

of the association rules at each database. McClean, Scotney, Greer and Pairceir (2001) consider the clustering of heterogeneous databases that hold aggregate count data.

They experimented with the Euclidean metric and the Kullback-Leibler information divergence for measuring the distance of aggregate data. Tsoumakas, Angelis and Vlahavas (2003) consider the clustering of databases in distributed classification tasks. They cluster the classification models that are produced at each site based on the differences of their predictions in a validation data set. Experimental results show that the combining of the classifiers within each cluster leads to better performance compared to combining all classifiers to produce a global model or using individual classifiers at each site.

TRENDS

One trend that can be noticed during the last years is the implementation of DDM systems using emerging distributed computing paradigms such as Web services and the application of DDM algorithms in emerging distributed environments, such as mobile networks, sensor networks, grids and peer-to-peer networks. Cannataro and Talia (2003), introduced a reference software architecture for knowledge discovery on top of computational grids, called Knowledge Grid. Datta, Bhaduri, Giannela, Kargupta and Wolff (2006), present an overview of DDM applications and algorithms for P2P environments. McConnell and Skillicorn (2005) present a distributed approach for prediction in sensor networks, while Davidson and Ravi (2005) present a distributed approach for data pre-processing in sensor networks.

CONCLUSIONS

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

REFERENCES

1. Agrawal R. & Srikant, R. (1994, September). Fast Algorithms for Mining Association Rules. In Proceedings of the 20th International Conference on Very Large Databases (VLDB'94), Santiago, Chile, 487-499.
2. Agrawal, R. & Shafer J.C. (1996) Parallel Mining of Association Rules. IEEE Transactions on Knowledge and Data Engineering, 8(6), 962-969.
3. Aronis, J. Kulluri, V., Provost, F., & Buchanan, B. (1997). The WoRLD: Knowledge discovery and multiple distributed databases. in Proceeding of florida artificial intellegence research symposium (FLAIRS-97).
4. Ashrafi, M.Z., Taniar, D. & Smith, K.(2004). ODAM: An Optimized Distributed Association Rule Mining Algorithm. IEEE Distributed Systems Online, 5(3).
5. Bauer, E., & Kohavi, R. (1999). An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. Machine Learning, 36 (1-2), 105-139.
6. Bhat, P.B., Raghavendra, C.S., Prasanna, V.K., "Efficient collective communication in distributed heterogeneous systems", Journal of Parallel and Distributed Computing 63 (2003), 251-263.

7. Breiman, L. (1999). Pasting small votes for classification in large databases and on-line. Machine Learning, 36 (1-2), 85-103.
8. Cannataro, M. and Talia, D. (2003). The Knowledge Grid. Communications of the ACM, 46(1), 89-93.
9. Cannataro, M., Talia, D., Runfio, P., "KNOWLEDGE GRID: High Performance Knowledge Discovery on the Grid", GRID 2001 (2001), 38-50.
10. 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.
11. Chan, P., & Stolfo, S. (1993a). Experiments on multi strategy learning by meta-learning. In Proceeding of the second international conference on information knowledge management (pp.314-323.).
12. Chan, P., & Stolfo, S. (1993b). Toward parallel and distributed learning by meta-learning. In Working notes aaai work knowledge discovery in database (pp.227-240). AAAI.
13. Chervanak, A., Foster, I., Kesslman, C., Salisbury, C., Tuecke, S., "The Data Grid: Towards an Architecture for the Distributed Management and Analysis of Large Scientific Datasets (1999).
14. Cheung, D.W., Han, J., Ng, V., Fu, A.W. & Fu, Y (1996, December) A Fast Distributed Algorithm for Mining Association Rules. In Proceedings of the 4th International Conference on Parallel and Distributed Information System (PDIS-96), Miami Beach, Florida, USA, 31-42.
15. Cheung, D.W., Ng, V., Fu, A.W. & Fu, Y. (1996). Efficient Mining of Association Rules in Distributed Databases. IEEE Transactions on Knowledge and Data Engineering, 8(6), 911-922.
16. Clifton, C., Marks, D., "Security and privacy implications of data mining", in of British Columbia Department of Computer Science, U., Ed.: In Workshop on Data Mining and Knowledge Discovery, Montreal, Canada (1996), 15C19.
17. Datta, S, Bhaduri, K., Giannella, C., Wolff, R. & Kargupta, H. (2006). Distributed Data Mining in Peer-to-Peer Networks, IEEE Internet Computing 10(4), 18-26.
18. Davidson I. & Ravi A. (2005). Distributed Pre-Processing of Data on Networks of Berkeley Motes Using Non-Parametric EM. In Proceedings of 1st International Workshop on Data Mining in Sensor Networks, 17-27.
19. 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.
20. 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.
21. Fan, W., Stolfo, S., & Zhang, J. (1999), The applciation of adaboost for distributed, scalable and on-line learning. In Fifth acm sigkdd international conference on knowledge discovery and data mining. San Diego, California.
22. Forman, G., & Zhang, B. (2000). Distributed data clustering can be efficient and exact. In Sigkdd exlorations (Vol.2).

23. 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 (IIP 2000). Beijing, China.

24. 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. Hillal Kargupa and Phillip Chan (pp.115-132) IT Press.

25. 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.

26. Kargupta, H., Hamzaoglu, I., Stafford, B., Hanagandi, V., & Buesher, 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.

27. Kargupta, H., Park, B.H., "A fourier spectrum-based approach to represent decision trees for mining data streams in mobile environments", IEEE Transactions on Knowledge & Data Eng. (2004).

28. 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.

29. Mastroianni, C., Talia, D., Trunfio, P., "Managing Heterogeneous Resources in Data Mining Applications on Grids Using XML-Based Metadata", In: IPDPS, Nice, France (2003).

30. McClean, S., Scotney, B., & Greer, K.(2000). Clustering heterogeneous distributed databases. In Workshop on distribted and parallel knowledge discovery. Boston, MA, USA.

31. McClean, S., Scotney, B., Greer, K. & P Páircéir, R.(2001). Conceptual Clustering of Heterogeneous Distributed Databases In Proceedings of the PKDD'01 Workshop on Ubiquitous Data Mining.

32. McConnell S. and Skillicorn D. (2005). A Distributed Approach for Prediction in Sensor Networks. In Proceedings of the 1st International Workshop on Data Mining in Sensor Networks, 28-37.

33. Merugu, S., Ghosh, J, "Privacy-preserving distributed clustering using generative models", in the Third IEEE International conference on Data Mining (ICDM'03), Melbourne, FL (2003).

34. Merz, C.J., & Pazzani, M.J. (1999). A principal components approach to combining regression estimates. Machine Learning, 36(1-2), 9-32.

35. Opitz, D., & Maclin, R. (1999). Popular ensemble methods: An empirical study: Journal of Artificial Intelligence Research, 11, 169-198.

36. Park, B., Kargupta, H., Johnson, E., Sanseverino, E., Hershberger, D., & Silverstre, L. (2002). Distributed, collaborative data analysis from heterogeneous sites using a scalable evolutionary technique. Applied Intelligence, 16(1).

37. Parthasarathy, S. & Ogihara, M. (2000). Clustering Distributed Homogeneous Databases. In Proceedings of the

4th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD-00), Lyon, France, September 13-16, 566-574.

38. Provost, F., "Distributed Data Mining: Scaling Up and Beyond,", In Kargupta, H., Chan, P., eds: Advances in Distributed Data Mining, MIT/AAAI Press (2000).
39. Provost, F.J., & Buchanan, B. (1995). Inductive policy: The pragmatics of bias selection. Machine Learning, 20, 35-61.
40. Samatova, N., Ostrouchov, G., Geist, A., & Melechko, A. (2002). Rachet: An efficient cover-based merging of clustering hierarchies from distributed datasets. An international Journal of Distributed and Parallel Databases, 11(2), 157-180.
41. Sayal, M., & Scheuermann, P. (2000). A distributed clustering algorithm for web-based access patterns. In Workshop on distributed and parallel knowledge discovery of KDD-2000 (pp.41-48). Boston.
42. Stolfo, S., et al., "JAM: Java Agents for Meta-learning over Distributed Databases", In: Proceedings of Third International Conference on Knowledge Discovery and Data Mining, Menlo Park, CA, AAA1 Press (1997) 74-81.
43. Ting & Low, B. (1997). Model combination in the multiple-data-base scenario. In 9th European conference on machine learning (pp.250-265.)
44. Tsoumakas, G., Angelis, L. & Vlahavas, I. (2003). Clustering Classifiers for Knowledge Discovery from Physically Distributed Databases. Data & Knowledge Engineering 49(3), 223-242.
45. 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.
46. Verykios, V.S., Bertino, E., Fovino, I.N., Provenza, L.P, Saygin, Y., Theodoridis, Y., "State-of-the-art in privacy preserving data mining", SIGMOD Record 33 (2004) 50-57.
47. Wolpert, D. (1992). Stacked generalization. Neural Networks, 5, 241-259.
48. 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.