Distributed Data Mining

Grigorios Tsoumakas*
Department of Informatics
Aristotle University of Thessaloniki
Thessaloniki, 54124
Greece
voice: +30 2310-998418
fax: +30 2310-998419
email: greg@csd.auth.gr

Ioannis Vlahavas
Department of Informatics
Aristotle University of Thessaloniki
Thessaloniki, 54124
Greece
voice: +30 2310-998418
fax: +30 2310-998419
email: vlahavas@csd.auth.gr

(* Corresponding author)

Distributed Data Mining

Grigorios Tsoumakas, Aristotle University of Thessaloniki, Greece

Ioannis Vlahavas, Aristotle University of Thessaloniki, Greece

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:

(1) *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 scale 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.

(2) *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.).

(3) *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.

(4) *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.

This article is concerned with Distributed Data Mining algorithms, methods and systems that deal with the above issues in order to discover knowledge from distributed data in an effective and efficient way.

BACKGROUND

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.

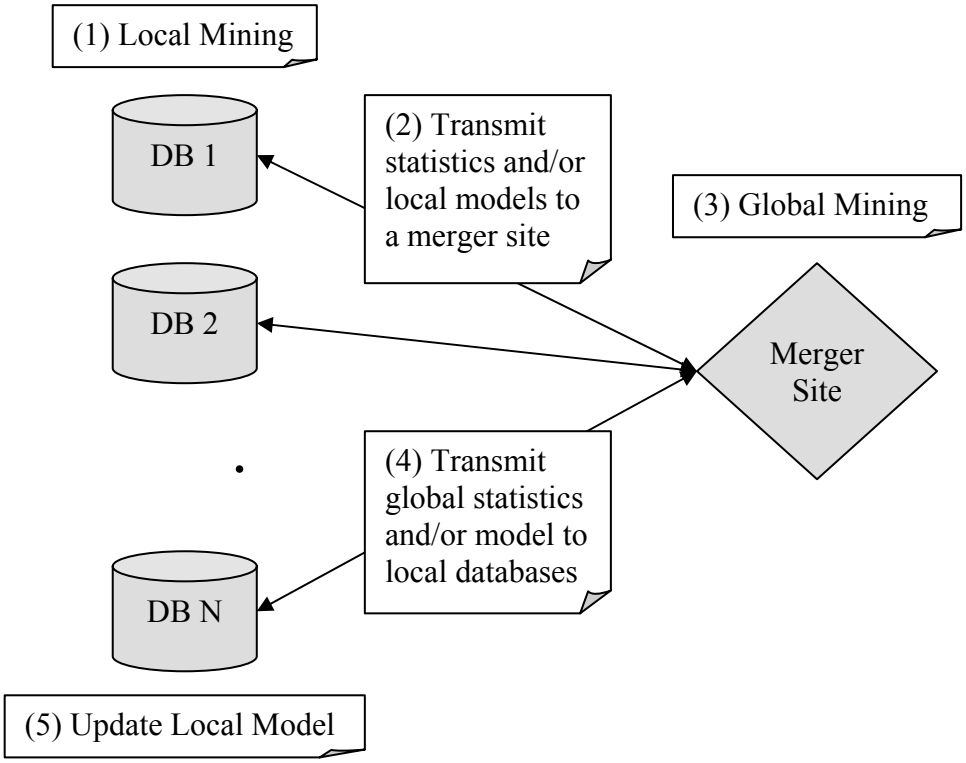

Figure 1: Typical architecture of Distributed Data Mining approaches

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.

MAIN FOCUS

Distributed Classification and Regression

Approaches for distributed classification and regression are mainly inspired from methods that appear in the area of ensemble methods, such as Stacking, Boosting, Voting and others. Some distributed approaches are straightforward adaptations of ensemble methods in a distributed computing environment, while others extend the existing approaches in order to minimize the communication and coordination costs that arise.

Chan and Stolfo (1993) applied the idea of Stacked Generalization (Wolpert, 1992) to DDM via their meta-learning methodology. They focused on combining distributed data sets and investigated various schemes for structuring the meta-level training examples. They showed that meta-learning exhibits better performance with respect to majority voting for a number of domains. Knowledge Probing (Guo & Sutiwaraphun, 1999) builds on the idea of meta-learning and in addition uses an independent data set, called the probing set, in order to discover a comprehensible model. The output of a meta-learning system on this independent data set together with the attribute value vector of the same data set are used as training examples for a learning algorithm that outputs a final model.

The Collective Data Mining (CDM) framework (Kargupta, Park, Hershberger & Johnson, 2000) allows the learning of classification and regression models over heterogeneous databases. It is based on the observation that any function can be represented using an appropriate set of basis functions. Initially, CDM generates approximate orthonormal basis coefficients at each site. It then moves an appropriately chosen sample of each data set to a single site and generates the approximate basis coefficients corresponding to non-linear cross terms. Finally, it

combines the local models and transforms the global model into the user-specified canonical representation.

A number of approaches have been presented for learning a single rule set from distributed data. Hall, Chawla and Bowyer (1997; 1998) present an approach that involves learning decision trees in parallel from disjoint data, converting trees to rules and then combining the rules into a single rule set. Hall, Chawla, Bowyer and Kegelmeyer (2000) present a similar approach for the same case, with the difference that rule learning algorithms are used locally. In both approaches, the rule combination step starts by taking the union of the distributed rule sets and continues by resolving any conflicts that arise. Cho and Wüthrich (2002) present a different approach that starts by learning a single rule for each class from each distributed site. Subsequently, the rules of each class are sorted according to a criterion that is a combination of confidence, support and deviation, and finally the top k rules are selected to form the final rule set. Conflicts that appear during the classification of new instances are resolved using the technique of relative deviation (Wüthrich, 1997).

Fan, Stolfo and Zhang (1999) present d-sampling AdaBoost, an extension to the generalized AdaBoost learning algorithm (Schapire and Singer, 1999) for DDM. At each round of the algorithm, a different site takes the role of training a weak model using the locally available examples weighted properly so as to become a distribution. Then, the update coefficient α_t is computed based on the examples of all distributed sites and the weights of all examples are updated. Experimental results show that the performance of the proposed algorithm is in most cases comparable to or better than learning a single classifier from the union of the distributed data sets, but only in certain cases comparable to boosting that single classifier. The distributed boosting algorithm of Lazarevic and Obradovic (2001) at each round learns a weak model in

each distributed site in parallel. These models are exchanged among the sites in order to form an ensemble, which takes the role of the hypothesis. Then, the local weight vectors are updated at each site and their sums are broadcasted to all distributed sites. This way each distributed site maintains a local version of the global distribution without the need of exchanging the complete weight vector. Experimental results show that the proposed algorithm achieved classification accuracy comparable or even slightly better than boosting on the union of the distributed data sets.

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 $O(|C_k|n^2)$, where C_k 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). DMA generates 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

Johnson and Kargupta (1999) present the Collective Hierarchical Clustering (CHC) algorithm for clustering distributed heterogeneous data sets, which share a common key attribute. CHC comprises three stages: i) local hierarchical clustering at

each site, ii) transmission of the local dendrograms to a facilitator site, and iii) generation of a global dendrogram. CHC estimates a lower and an upper bound for the distance between any two given data points, based on the information of the local dendrograms. It then clusters the data points using a function on these bounds (e.g. average) as a distance metric. The resulting global dendrogram is an approximation of the dendrogram that would be produced if all data were gathered at a single site.

Samatova, Ostrouchov, Geist and Melechko (2002), present the RACHET algorithm for clustering distributed homogeneous data sets. RACHET applies a hierarchical clustering algorithm locally at each site. For each cluster in the hierarchy it maintains a set of descriptive statistics, which form a condensed summary of the data points in the cluster. The local dendrograms along with the descriptive statistics are transmitted to a merging site, which agglomerates them in order to construct the final global dendrogram. Experimental results show that RACHET achieves good quality of clustering compared to a centralized hierarchical clustering algorithm, with minimal communication cost.

Januzaj, Kriegel and Pfeifle (2004) present the Density Based Distributed Clustering (DBDC) algorithm. Initially, DBDC uses the DBSCAN clustering algorithm locally at each distributed site. Subsequently, a small number of representative points that accurately describe each local cluster are selected. Finally, DBDC applies the DBSCAN algorithm on the representative points in order to produce the global clustering model.

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 Páircéir (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.

FUTURE 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.

CONCLUSION

DDM enables learning over huge volumes of data that are situated at different geographical locations. It supports several interesting applications, ranging from fraud and intrusion detection, to market basket analysis over a wide area, to knowledge discovery from remote sensing data around the globe.

As the network is increasingly becoming the computer, the role of DDM algorithms and systems will continue to play an important role. New distributed applications will arise in the near future and DDM will be challenged to provide robust analytics solutions for these applications.

REFERENCES

Agrawal, R. & Shafer J.C. (1996). Parallel Mining of Association Rules. *IEEE Transactions on Knowledge and Data Engineering*, *8*(6), 962-969.

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.

Ashrafi, M. Z., Taniar, D. & Smith, K. (2004). ODAM: An Optimized Distributed Association Rule Mining Algorithm. *IEEE Distributed Systems Online*, *5*(3).

Cannataro, M. and Talia, D. (2003). The Knowledge Grid. *Communications of the ACM, 46*(1), 89-93.

Chan, P. & Stolfo, S. (1993). *Toward parallel and distributed learning by meta-learning*. In Proceedings of AAAI Workshop on Knowledge Discovery in Databases, 227-240.

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.

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.

Cho, V. & Wüthrich, B. (2002). Distributed Mining of Classification Rules. *Knowledge and Information Systems, 4,* 1-30.

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.

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.

Fan, W., Stolfo, S. & Zhang, J. (1999, August). *The Application of AdaBoost for Distributed, Scalable and On-Line Learning*. In Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, California, USA, 362-366.

Fu, Y. (2001). Distributed Data Mining: An Overview. *Newsletter of the IEEE Technical Committee on Distributed Processing*, Spring 2001, pp.5-9.

Guo, Y. & Sutiwaraphun, J. (1999). *Probing Knowledge in Distributed Data Mining*. In Proceedings of the Third Pacific-Asia Conference on Methodologies for Knowledge Discovery and Data Mining (PAKDD-99), 443-452.

Hall, L.O., Chawla, N., Bowyer, K. & Kegelmeyer, W.P. (2000). Learning Rules from Distributed Data. In M. Zaki & C. Ho (Eds.), *Large-Scale Parallel Data Mining*. (pp. 211-220). LNCS 1759, Springer.

Hall, L.O., Chawla, N. & Bowyer, K. (1998, July). *Decision Tree Learning on Very Large Data Sets*. In Proceedings of the IEEE Conference on Systems, Man and Cybernetics.

Hall, L.O., Chawla, N. & Bowyer, K. (1997). *Combining Decision Trees Learned in Parallel*. In Proceedings of the Workshop on Distributed Data Mining of the 4th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

Johnson, E.L & Kargupta, H. (1999). Collective Hierarchical Clustering from Distributed, Heterogeneous Data. In M. Zaki & C. Ho (Eds.), *Large-Scale Parallel Data Mining*. (pp. 221-244). LNCS 1759, Springer.

Kargupta, H., Park, B-H, Herschberger, D., Johnson, E. (2000) Collective Data Mining: A New Perspective Toward Distributed Data Mining. In H. Kargupta & P. Chan (Eds.), *Advances in Distributed and Parallel Knowledge Discovery*. (pp. 133-184). AAAI Press.

Lazarevic, A, & Obradovic, Z. (2001, August). The Distributed Boosting Algorithm. In Proceedings of the 7th ACM SIGKDD international conference on Knowledge discovery and data mining, San Francisco, California, USA, 311-316.

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.

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.

Park, B. & Kargupta, H. (2003). Distributed Data Mining: Algorithms, Systems, and Applications. In N. Ye (Ed.), *The Handbook of Data Mining*. (pp. 341-358). Lawrence Erlbaum Associates.

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.

Samatova, N.F., Ostrouchov, G., Geist, A. & Melechko A.V. (2002). RACHET: An Efficient Cover-Based Merging of Clustering Hierarchies from Distributed Datasets. *Distributed and Parallel Databases 11*, 157-180.

Schapire, R & Singer, Y. (1999). Improved boosting algorithms using confidence-rated predictions. *Machine Learning 37*(3), 297-336.

Tsoumakas, G., Angelis, L. & Vlahavas, I. (2003). Clustering Classifiers for Knowledge Discovery from Physically Distributed Databases. *Data & Knowledge Engineering 49*(3), 223-242.

Wolpert, D. (1992). Stacked Generalization. *Neural Networks 5*, 241-259.

Wüthrich, B. (1997). Discovery probabilistic decision rules. International Journal of Information Systems in Accounting, Finance, and Management 6, 269-277.

KEY TERMS AND THEIR DEFINITIONS

Data Skewness: The observation that the probability distribution of the same attributes in distributed databases is often very different.

Distributed Data Mining (DDM): A research area that is concerned with the development of efficient algorithms and systems for knowledge discovery in distributed computing environments.

Global Mining: The combination of the local models and/or sufficient statistics in order to produce the global model that corresponds to all distributed data.

Grid: A network of computer systems that share resources in order to provide a high performance computing platform**.**

Homogeneous and Heterogeneous Databases: The schemata of Homogeneous (Heterogeneous) databases contain the same (different) attributes.

Local Mining: The application of data mining algorithms at the local data of each distributed site.

Sensor Network: A network of spatially distributed devices that use sensors in order to monitor environment conditions.