Volume 4, Issue 11, November 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

A Novel Based Scattered Multi-Agent and Data Mining

M. Prakash, N. Parasuraman
Assistant Professor Department of Computer Science
Shanmuga Industries Arts & Science College, Manalurpettai Road,
Tiruvannamalai, India

Abstract: In this paper focuses on the classification problem in distributed data mining environments where the transfer of data between learning processes is limited. Existing solutions address this problem through the use of distributed technologies for applying data mining algorithms to learn global models from local learning processes. Multiagent based solutions that follow this approach overlook the autonomy of local learning processes, the decentralisation of system control, and the local learning heterogeneity of the processes. We propose a collaborative agent-based learning model inspired by an existing learning framework that overcomes these deficiencies by defining the overall learning process as a combination of local autonomous learners interacting with each other in order to improve their local classification performance. Our model is an extension of this work and redefines agent learning behaviour as consisting of four distinct steps: the selection of the learner with which to interact, the integration of acquired knowledge, the evaluation of the resulting model and the update of the learning knowledge. For each of these different steps, several methods and criteria have been proposed in order to offer different alternatives for configuring the collaborative learning algorithm for limited data sharing domains.

Keywords: distributed data mining, NOC, multi-agent, multi-database, multi-relational mining, game theory.

I. INTRODUCTION

Data mining technology has emerged as a means for identifying patterns and trends from large quantities of data. Distributed Data Mining (DDM) aims at extraction useful pattern from distributed heterogeneous data bases in order, for example, to compose them within a distributed knowledge base and use for the purposes of decision making. A lot of modern applications fall into the category of systems that need DDM supporting distributed decision making. Applications can be of different natures and from different scopes, for example, data and information fusion for situational awareness; scientific data mining in order to compose the results of diverse experiments and design a model of a phenomena, intrusion detection, analysis, prognosis and handling of natural and man-caused isaster to prevent their catastrophic development, Web mining ,etc. From practical point of view, DDM is of great concern and ultimate urgency. A network operations center (or NOC, pronounced "knock") is one or more locations from which control is exercised over a computer, television broadcast, or telecommunications network. Large organizations may operate more than one NOC, either to manage different networks or to provide geographic redundancy in the event of one site being unavailable or offline. NOCs are responsible for monitoring the network for alarms or certain conditions that may require special attention to avoid impact on the network's performance. For example, n a telecommunications environment, NOCs are responsible for monitoring for power failures, communication line alarms (such as bit errors, framing errors, line coding errors, and circuits down) and other performance issues that may affect the network.

Our approach takes a step toward solving distributed data mining for the classification problem and proposes an alternative based on using communication and collaboration among the different local classification learning processes. More specifically, the solution adopted makes use of the multiagent system paradigm and redefines the learning processes by viewing it as a group of autonomous, heterogeneous and collaborative learning agents. Our solution envisages the system as a society of learning agents with communication and reasoning capabilities that interact among themselves in a decentralised fashion.

Many data mining applications, both current and proposed are faced with an active adversary. Problems range from the annoyance of spam to the damage of computer hackers to the destruction of terrorists. In all of these cases, statistical classification techniques play an important role in distinguishing the legitimate from the destructive. There has been significant investment in the use of learned classifiers to address these issues, from commercial spam filters to research programs such as those on intrusion detection. These problems pose a significant new challenge not addressed in previous research: The behavior of a class (the adversary) may adapt to avoid detection. A classifier constructed by the data miner in a static environment won't maintain its optimal performance for long, when facing an active adversary.

An intuitive approach to fight the adversary is to let the classifier adapt to the adversary's actions, either manually or automatically. Such a classifier was proposed in [1], which left open the following issue. The problem is that this becomes a never-ending game between the classifier and the adversary. Or is it never-ending? Will we instead reach an equilibrium, where each party is doing the best it can and has no incentive to deviate from its current strategy? If so, does this equilibrium give a satisfactory result for those using the classifier? Or does the adversary win? Our approach is *not* to

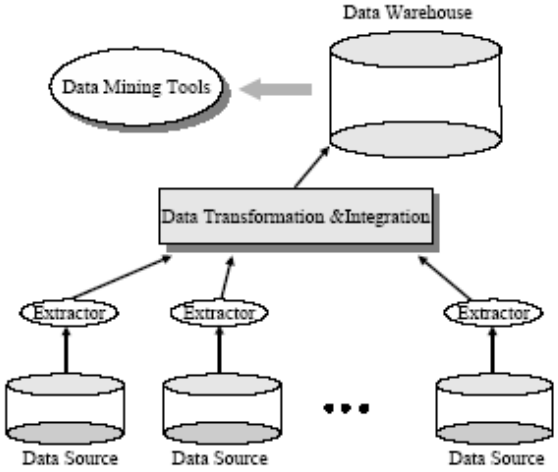
develop a learning strategy for the classifier to stay ahead of the adversary. We instead predict the end state of the "game"- an equilibrium state. We model the problem as a two-player game, where the adversary tries to maximize its return and the data miner tries to minimize the amount of misclassification. We examine under which conditions an equilibrium would exist, and provide a method to estimate the classifier performance and the adversary's behavior at such an equilibrium point (e.g., the players' equilibrium strategies). Spam filtering is one motivating application.

There are many examples of spam e-mails where words are modified to avoid spam filters. We could see that those transformations the adversary makes to defeat the data miner come with a cost: lower response rates. Combining the fact that the reward to the adversary decreases as they try to defeat the data miner, with the data miner's interest in avoiding false positives as well as false negatives, can lead us to equilibrium where both are best served by maintaining the status quo.

A game is a formal description of a strategic situation. Game theory is the formal study of decision-making where several players must make choices that potentially affect the interests of the other players. The remaining sections of the paper are organized as follows. In Section II we describe the distributed data mining. In Section III we describe MultiData base Mining In Section IV we describe Agent-based distributed data mining and open problems Strategy Section V A game Theoretic Model Section VI concludes the paper.

II. DISTRIBUTED DATA MINING

Data mining technology has emerged as a means for identifying patterns and trends from large quantities of data. Data mining and data warehousing go hand-in-hand: most tools operate on a principal of gathering all data into a central site, then running an algorithm against that data (Figure 1). There are a number of applications that are infeasible under such a methodology, leading to a need for distributed data mining.



III. MULTI DATA BASE MINING

Business, government and academic sectors have all implemented measures to computerize all, or part of, their daily functions [9]. An interstate (or international) company consists of multiple branches. The National Bank of Australia, for example, has many branches in different locations. Each branch has its own database, and the bank data is widely distributed and thus becomes a multi-database.

Functional design of the application

The application should permit the execution of several experiments over different environment configurations in order to exhaustively test our learning model. For this reason, we have designed an application where we first specify the environment parameters and the learning experiments to be run in these environments, then the application executes these experiments, and finally summary results of these runs are produced.

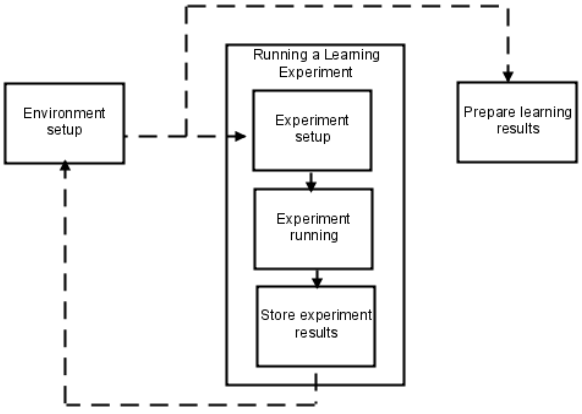


Fig2: Execution flow of the application

A variety of different factors can affect the performance of our collaborative learning model, some of which depend on the environment in which it is deployed, like the number of learners in the system, the learning algorithms used, or the size of the datasets used for training the learners. Other factors depend on the internal configuration of collaborative learning itself, such as the neighbour selection criterion chosen, the knowledge integration method employed, and so on.
To account for these factors, we developed an application that permits the testing of the collaborative learning model using different parametrisations of the environment or of the learning components themselves. In the following sections, we provide the details of the design and implementation of this application.
– The use of a greedy accuracy selection tactic versus a randomised accuracy weighted strategy.
– The use of methods for environments in which transfer of small amounts of data is allowed.
– The use of methods for environments in which classification outputs may be communicated.
– The use of methods for environments in which local models may be communicated.
– The effect of allowing more interactions between the learning agents.

VI. IMPLEMENTATION OF THE APPLICATION

The application has been implemented using Java technology because, among other advantages, it offers object oriented technology; it is broadly used and lots of pre-existing libraries are available; and, it offers an easy environment for programming. Also, we have used the open source library for machine learning tasks, such as classifier building, training and evaluation. We have chosen Weka because it offers three principal advantages over most other data mining software packages. Firstly, it is open source, which not only means that it can be obtained for free, but it is maintainable, and modifiable. Secondly, it provides a number of state-of-the-art machine learning algorithms that can be deployed in any given problem. Thirdly, it is implemented in Java and hence fully compatible with the implementation of our application.

V. CONCLUSION

This paper has shown that the problem of distributed data mining and mining multi-database is challenging and pressing. We have defined a new process of multidatabase mining for our system with game theory. In domains ranging from spam detection to counterterrorism, classifiers have to contend with adversaries manipulating the data to produce false negatives. Research in this direction has the potential to produce DDM systems that are more robust to adversary manipulations and require less human intervention to keep up with them. Many classification problems operate in a setting with active adversaries: while one party tries to identify the members of a particular class, the other tries to reduce the effectiveness of the classifier. Although this may seem like a never-ending cycle, it is possible to reach a steady-state where the actions of both parties stabilize. The game has equilibrium because both parties facing costs: costs associated with misclassification on the one hand, and for defeating the classifier on the other. By incorporating such costs in modeling, we can determine where such equilibrium could be reached, and whether it is acceptable to the data miner.

REFERENCES

[1] X.Wu and S. Zhang, Synthesizing High-Frequency Rules from Different Data Sources, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 15, No. 2, March/April 2003: 353-367.
[2] C. Zhang and S. Zhang, *Association Rules Mining: Models and Algorithms*. Springer- Verlag Publishers in Lecture Notes on Computer Science, Volume 2307, p. 243, 2002.
[3] N. Zhong, Y. Yao, and S. Ohsuga, Peculiarity oriented multi-database mining. In: Proceedings of PKDD, 1999: 136-146.
[4] Date, C. *An Introduction to Database Systems, Volume I*, The Systems Programming Series, Addison-Wesley, 1986.
[5] Ullman, I.D. *Principles of Databases and Knowledge-Based Systems,*Volume I, Computer Science Press, 1988
[6] Ullman, J., Widom, J., *A First Course in Database Systems*, Prentice Hall, 2001
[7] R. B. Doorenbos, O. Etzioni, and D. S. Weld. A scalable comparison-shopping agent for the World-Wide Web. In Proceedings of the First International Conference on Autonomous Agents, pages 39{48, Marina del Rey, CA, 1997. ACM Press.
[8] T. Fawcett. \In vivo" spam _ltering: A challenge problem for KDD. SIGKDD Explorations, 5(2):140{148, 2003. T. Fawcett and F. Provost. Adaptive fraud detection. Data Mining and Knowledge Discovery, 1(3):291{316,
[9] L. Guernsey. Retailers rise in Google rankings as rivals cry foul. New York Times, November 20, 2003.
[10] D. Jensen, M. Rattigan, and H. Blau. Information awareness: A prospective technical assessment. In Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 378{387, Washington, DC, 2003. ACM Press.
[11] B. Krebs. Online piracy spurs high-tech arms race.Washington Post, June 26, 2003.

