Agents and Data Mining: Mutual Enhancement by Integration

Chengqi Zhang[1], Zili Zhang[2,3], and Longbing Cao[1]

[1] Faculty of Information Technology
University of Technology, Sydney
PO Box 123, Broadway NSW 2007, Australia
{chengqi, lbcao}@it.uts.edu.au
[2] Faculty of Computer and Information Science
Southwest China Normal University, Chongqing 400715, China
zhangzl@swnu.edu.cn
[3] School of Information Technology
Deakin University, Geelong, VIC 3217, Australia
zzhang@deakin.edu.au

Abstract. This paper tells a story of synergism of two cutting edge technologies - agents and data mining. By integrating these two technologies, the power for each of them is enhanced. Integrating agents into data mining systems, or constructing data mining systems from agent perspectives, the flexibility of data mining systems can be greatly improved. New data mining techniques can add to the systems dynamically in the form of agents, while the out-of-date ones can also be deleted from systems at run-time. Equipping agents with data mining capabilities, the agents are much smarter and more adaptable. In this way, the performance of these agent systems can be improved. A new way to integrate these two techniques –ontology-based integration is also discussed. Case studies will be given to demonstrate such mutual enhancement.

1 Introduction

Agents (adaptive or intelligent agents and multi-agent systems) constitute one of the most prominent and attractive technologies in Computer Science at the beginning of this new century. Agent and multi-agent system technologies, methods, and theories are currently contributing to many diverse domains. These include information retrieval, user interface design, robotics, electronic commerce, computer mediated collaboration, computer games, education and training, smart environments, ubiquitous computers, and social simulation.

This is not only a very promising technology, it is emerging as a new way of thinking, a conceptual paradigm for analyzing problems and for designing systems, for dealing with complexity, distribution and interactivity, and perhaps a new perspective on computing and intelligence.

Agent-based computing has been a source of technologies to a number of research areas, both theoretical and applied. These include distributed planning and decision-making, automated auction mechanisms and learning mechanisms. Moreover, agent technologies have drawn from, and contributed to, a diverse range of academic disciplines, in the humanities, the sciences and social sciences. The fundamental research issues in agent technologies include multi-agent planning, agent communication languages, coordination mechanisms, matchmaking architectures and algorithms, information agents and basic ontologies, sophisticated auction mechanism design, negotiation strategies, and learning [8].

Agent technologies are a natural extension of current component-based approaches, and have the potential to greatly impact the lives and work of all of us and, accordingly, this area is one of the most dynamic and exciting in computer science today [9].

Data mining (also known as Knowledge Discovery in Databases - KDD) has been defined as "The nontrivial extraction of implicit, previously unknown, and potentially useful information from data." [10]. Using a combination of machine learning, statistical analysis, modeling techniques and database technology, data mining finds patterns and subtle relationships in data and infers rules that allow the prediction of future results. Typical applications include market segmentation, customer profiling, fraud detection, evaluation of retail promotions, and credit risk analysis.

Data mining is a powerful new technology with great potential to help companies focus on the most important information in their data warehouses. Data mining tools predict future trends and behaviors, allowing businesses to make proactive, knowledge-driven decisions. The automated, prospective analyses offered by data mining move beyond the analyses of past events provided by retrospective tools typical of decision support systems. Data mining tools can answer business questions that traditionally were too time consuming to resolve. They scour databases for hidden patterns, finding predictive information that experts may miss because it lies outside their expectations.

The large variety of data mining techniques which have been developed over the past decade includes methods for pattern-based similarity search, cluster analysis, decision-tree based classification, generalization taking the data cube or attribute-oriented induction approach, and mining of association rules [11].

Recent years saw a new trend in the combination of the multi-agent system approach, data mining and KDD. It could be observed that recent research areas in multi-agent systems (agent behavior adaptation and reinforcement learning, collective behavior learning, e.g., through modification of collective behavior protocols, etc.) utilize data mining and KDD as a source of innovative ideas. In return, multi-agent technology is used to address architectural and implementation issues in engineering and software implementation of data mining and KDD systems.

In this paper, we will tell a story of synergism of agent and data mining technologies based on our research as well as other peer colleagues. In Section 2, how agent technology can be used to facilitate the construction data mining systems is discussed. Section 3 presents how to enhance the capabilities of agents by equipping data mining ability. Section 4 demonstrates ontology-based integration of agents and data mining – a new way to integrate these two cutting edge technologies. Finally, Section 5 concludes this paper.

2 Agent-Based Data Mining

Most of the existing data mining techniques were originally developed for centralized data and need to be modified for handling the distributed case. This motivated the development of parallel and distributed data mining. However, current techniques developed for parallel/distributed data mining failed to meet the requirements of the problems. The inherent feature of agents of being autonomous, capable of adaptive and deliberative reasoning seems to fit quite well with the requirements of coping with the challenges of distributed data mining. Agent techniques play an important role in distributed data mining. Some typical work in this area is summarized in Section 2.1.

On the other hand, data mining systems are typical complex systems and difficult to construct due to many techniques and iterative steps involved in the process [12][13][14]. An agent is a computer system capable of flexible autonomous action in a dynamic, unpredictable and open environment. Agents offer a new and often more appropriate route to the development of complex systems, especially in open and dynamic environments [9]. Thus agent approach is particularly well-suited for data mining construction. Our work related to this topic is reported in Section 2.2.

2.1 Agents in Distributed Data Mining

Autonomous data mining agents as a special kind of information agents may perform various kinds of mining operations on behalf of its user(s) or in collaboration with other agents. Systems of cooperative information agents for data mining tasks in distributed, heterogeneous and massive data environments appear to be quite a natural vision for the near future to be realized. In [1], Klusch et al. discussed the advantages by using agents for distributed data mining. They argued that employing agent techniques to build distributed data mining systems can bring the following benefits:

- *Remaining the autonomy of data sources.* A data mining agent can be considered as a modular extension of a data management system to deliberatively handle the access to the underlying data source in accordance with given constraints on the required autonomy of the system, data, and model. This is in full compliance with the paradigm of cooperative information systems.
- *Facilitating interactive distributed data mining.* Pro-actively assisting agents can drastically limit the amount a user has to supervise and interfere with the running data mining process. For example, data mining agents can anticipate the individual limits of the potentially large search space and proper intermediate results particularly driven by their individual users' preferences with respect to the particular type of data mining task at hand.

- *Improving dynamic selection of sources and data gathering.* One challenge for data mining systems used in open distributed data environments is to discover and select relevant sources. In such settings data mining agents can be applied to adaptively select data sources according to given criteria such as the expected amount, type and quality of data at the considered source, actual network and data mining server load. Such data mining agents can be used, for example, to dynamically control and manage the process of data gathering to support any online analytical processing and business data warehouse application.
- *Having high scalability to massive distributed data.* One option to reduce network and data mining application server load is to let data mining agents migrate to each of the local data sites in a distributed data mining system on which they can perform mining tasks locally, and then either return with or send relevant pre-selected data to their originating server for further processing.
- *Stimulating multi-strategy distributed data mining.* For some complex application settings and appropriate combination of multiple data mining techniques can be more beneficial than applying just one particular one. Data mining agents can learn in due course of their deliberative actions which one to choose depending on the type of data retrieved from different sites and mining tasks to be pursued.
- *Enabling collaborative data mining.* Data mining agents can operate independently on data they have gathered at local sites, and then combine their respective models. Or they can agree to share potential knowledge as it is discovered, in order to benefit from additional opinions of other data mining agents

Recently agent techniques have been applied to distributed data mining. The most prominent and representative agent-based distributed data mining systems include BODHI, PADMA, JAM, and Papyrus.

In [16] and [17], 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. In [18], 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 [19] 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].

2.2 Building Data Mining Systems from Agent Perspectives

It is well known that there are variety of methods related to different main tasks in data mining, and outputs of different data mining methods also have different forms. However, a single data mining technique has not been proven appropriate for every domain and data set. Instead, several techniques may need to be integrated into hybrid systems and used cooperatively during a particular data mining operation. Therefore, hybrid intelligent systems are required for data mining tasks. For further justifying this statement, a simple example is now provided.

The example is about how to identify a set of "promising" securities to be included in an investment portfolio based on the historical fundamental and technical data about securities. This is a very appropriate domain for data mining for two reasons. First, because the number of available securities being traded in the various exchanges is very large. Identifying appropriate securities for the goals of a particular portfolio is based on the close examination of the performance of these securities. Without the use of data mining techniques, analysts can only closely examine small amounts of such data. Second, analysts are able to state criteria for identifying securities that can potentially meet a set of investment goals. However, they cannot identify all the necessary criteria. Furthermore, even after a set of securities is identified, large volumes of data relating to these securities still has to be examined in order to fine-tune the stated performance criteria, as well as identify others not previously considered by the analyst. For this simple task, no single data mining technique is adequate. Methods to formulate a pattern (hypothesis) and test its validity on the target databases are needed. Methods to discover other relevant patterns from target databases are also required. Some other methods including classification method to classify each security, inductive learning methods, and visualization techniques are also helpful for this task. If we construct a computer system to perform this task, it is evident that this system is a hybrid system integrated different techniques.

Once again, data mining is iterative sequence of many steps, while many techniques are involved in each step. These techniques need to be integrated into hybrid systems and used cooperatively for data mining tasks.

We have argued that agent perspectives are well-suited to hybrid intelligent systems construction, and proposed an agent-based framework for complex problem solving [20]. This framework was successfully applied to build a hybrid intelligent system for data mining [4]. The data mining systems built from agent perspectives have the following crucial characteristics that differentiate from others:

– New data mining techniques can be added to the system and out-of-date techniques can be deleted from the system dynamically;
– Data mining related agents and other agents can interact at run-time with ease under this framework, but in other non-agent based systems, these interactions must be determined at design-time.

For demonstration purpose, we show the Weka system [15] re-implemented from agent perspectives. The main focus of Weka is on classifier and filter algorithms. It also includes implementations of algorithms for learning association rules and for clustering data for which no class value is specified.

To re-implement the programs in Weka from agent perspectives, the programs in Weka (written in Java) were compiled into .DLLs (dynamic link library) first. The Java Native methods and JATLite KQML layer templates were then employed to wrap these programs in .DLL. In this way, all the programs in Weka were equipped with KQML communication capability and are ready to add to the agent system.

In this agent-based data mining experimental system, in addition to the supporting agents (interface agent, planning agent, middle agent, and so on) there are 7 attribute selection related agents, 25 classifier related agents, 9 filter related agents, and 2 cluster related agents.

Figure 1 shows the user interface of the system, which can start from any Internet Browser or appletviewer.

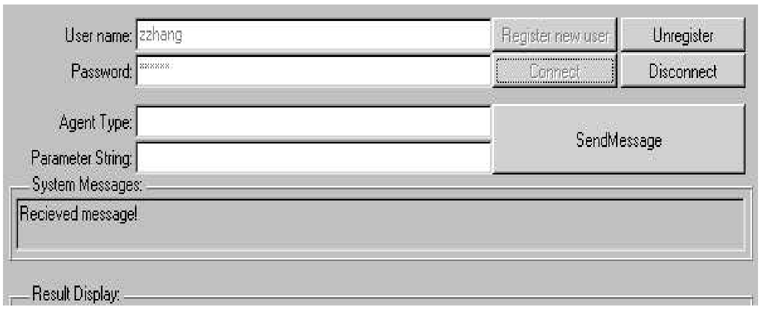

Fig. 1. User Interface of the System

To use the system, the user needs to type the user name and the password he likes in the corresponding fields and click “register new user” to register for the first time. Thereafter, just input the registered user name and password and click “connect”. If the user wants to leave the system, click “disconnect” and “unregister”.

The system can work in two modes. In one mode, all the data mining related agents can run individually, which is similar to execute the original Weka programs from the command line. In this mode, the user provides the system with the "agent type" and corresponding "parameter string" information in the corresponding input fields, and then click "SendMessage" button. The system will activate the given agent and display the results in the "result display" window. For example, if we type in "weke.classifiers.m5.M5Prime" in the "agent type" field, and "-t data\cpu.arff" in the "parameter string" field ("data\" was added before the data file as all data files in the system is in the "data" subdirectory), the system will display the following results (Figure 2), which are the same as running this program from the command line in Weka.

Fig. 2. Output from the M5 agent for numeric prediction

Another mode is to provide the planning agent with a work plan. The system then activates different agents based on the work plan. The work plan can be generated automatically based on meta-knowledge of the task in the planning agent. A work plan can also be edited manually according to the work plan specification and loaded into planning agent. Currently, only the latter implemented in the experimental system.

This example and other experimental systems we implemented indicate that flexible and robust data mining systems can be constructed with ease under the unified agent framework.

3 Agent Systems with Data Mining Capabilities

In this section we present an example to demonstrate that equipping agent systems with data mining abilities, the performance of agent systems can be improved. In the agent-based system for portfolio selection [7], data mining techniques were integrated.

Before we discuss the integration, we describe the process of analyzing large set of security data to discover the "promising" securities.

Assume the target database the analysis based contains monthly data of 2000 securities over a period of ten years.

Our goal is trying to create a "promising" security set of $15 - 20$ securities which the portfolio selection models will use (and also the investor will subsequently trade). Such data may be purchased from stock exchange Inc. such as Australian Stock Exchange Limited or downloaded from the corresponding Websites. Suppose we are interested in stocks with high earnings-per-share growth and high dividend growth, where "high" in both instances means greater than 50%. We define the concept of *strong security* as one whose *earning-per-share-growth* and *dividend-growth* is greater than 50%. We define "promising" security as one its *return on investment* two quarters later will be greater than 5%.

Next, we need to formulate a pattern (hypothesis) and test its validity on the target database. The hypothesis states that "IF a security is *strong* at time t, THEN its *return on investment* two quarters later will be greater than 5%." In addition to selecting the concepts that will be included in this pattern, we must also specify the bindings between the attributes of the selected concepts.

Once we instruct the system to validate the hypothesis, the system returns two sets of answers based on randomly selected sample data set: the records (securities) that support the hypothesis and those that refute it.

We then test the discovered patterns by applying them to another data set, called the *test set*, that was not originally used to discover the patterns. With each security to be classified the system returns its prediction, along with the confidence level associated with the prediction, as well as the actual return on investment.

By examining the evidence for incorrect predictions, we can identify rules that are suspect. We may iteratively apply top-down and bottom-up data mining to obtain more correct and complete models and higher prediction accuracy.

The collection of patterns created through the pattern validation and data exploration operations forms the rule-based model. This model is then applied on the set of securities that are targeted. The top 20 of the securities whose return on investment is predicted by the model to be "high", i.e., greater than 5% over a six month period are included in the "promising" security set.

In the whole process to determine the "promising" security set from large volumes of data, different data mining techniques such supervised symbolic inductive learning etc. are used.

By incorporating data mining techniques into the system, such a system can closely examine far more securities than manual approaches when creating "promising" security set. This laid a sound foundations for finding good portfolios.

Combining agent and data mining these two cutting edge technologies together can improve the performance of portfolio selection. Such agent-based portfolio selection system with data mining ability is very promising to find portfolios which best meet the objectives of the investors.

4 Ontology-Based Integration of Agents and Data Mining

From the discussions of previous two sections, it is evident that agent technology and data mining technology need each other. These two cutting edge technologies can enhance each other significantly by integration. In this section, we discuss an effective and efficient way to integrate agent and data mining, which is called *ontology-based integration*.

The key point in ontology-based integration is a view of ontological engineering in agent-based system analysis and design. The ontological engineering normally deals with the following processes [21]:

- building ontology profiles for the domain problem,
- defining ontological semantic relationships,
- representing ontologies, and
- aggregating and transforming ontologies in one or cross-domains. This actually is the foundation for system analysis, design and implementation.

The application process of ontological engineering is as follows.

- Understanding business ontology and user personalization in terms of business and user profiles.
- Extracting domain ontologies and problem-solving ontologies in terms of business processes, work flow, requirements engineering, and agent-oriented methodology.
- Abstracting semantic relationships among ontology concepts for different ontological domains.
- Representing ontologies visually and formally in terms of semantic relationships, ontological domains, and implementation techniques.
- Defining aggregation and transformation rules for ontology concepts, which support semantic aggregation, transformation and mapping between ontology concepts and key words, from one ontological item to another, or cross ontological domains.

According to the above process, an agent-based data mining infrastructure called F-TRADE (Financial Trading Rules Automated Development and Evaluation) was developed. F-TRADE is a web-based automated enterprise infrastructure for evaluation of trading strategies and data mining algorithms with online connection to huge amount of stock data. The current version F-TRADE 2.0 can be accessed at *http://datamining.it.uts.edu.au:8080/tsap*.

F-TRADE can provide financial traders and researchers, and miners on financial data with a practically flexible and automatic infrastructure. With this infrastructure, they can plug their algorithms into it easily, and concentrate on improving the performance of their algorithms with iterative evaluation on a large amount of real stock data from international markets. All other work, including user interface implementation, data preparation, and result output etc. is maintained by this platform. For financial traders, for instance, brokers and retailers, the F-TRADE presents them a real test bed, which can help them evaluate their favorite trading strategies iteratively without risk before they put money into the real markets. On the other hand, the F-TRADE presents a large amount of real stock data in multiple international markets, which can be used for both realistic back-testing of trading strategies and mining algorithms.

The F-TRADE looks also like an online service provider. As a systematic infrastructure for supporting data mining, trading evaluation, and finance-oriented applications, the F-TRADE encompasses comprehensive functions and services. They can be divided into the following groups: trading service support, mining services support, data services support, algorithm services support, and system services support. From the ontological engineering perspective, there are two key functions in F-TRADE that support all these services. One is soft plug-and-play [22], the other is ontology transformation in cross-domains [23].

Soft plug-and-play is essential in F-TRADE. It gets involved in plug in of data sources, data requests, trading or mining algorithms, system functional components, and the like. As a matter of fact, it has been a significant feature which supports the evolution of the F-TRADE and the application add-ons on top of the F-TRADE. The analysis, design, and implementation of soft plug-and-play are mainly supported by the agent service-oriented technique, which includes the analysis and design of the role model and the agent services for the plug-and-play. Refer to [22] for the details.

There are multiple heterogeneous ontological domains existing in F-TRADE. Flexible and efficient transformation, mapping and discovery of ontologies from multiple heterogeneous ontological domains are essential for the success of such systems.

The ontology processing engine in F-TRADE semantically aggregates and transforms user-defined business-oriented (or domain-oriented) special terms to internal standard items used by the problem-solving system. Three aspects must be followed in order to do the semantic aggregation and ontological transformation from user-defined keywords to ontological elements in problem-solving ontological domain. They are (i) semantic aggregation between semantic relationships, (ii) semantic aggregation of ontological items, and (iii) transformation from one ontology domain to another. All the above three types of transformations can happen in either one ontological domain or multiple domains.

Semantic Aggregation of relationships is to study whether there are rules for transitivity, additivity and anti-symmetry which can be performed between ontological semantic relationships. The aggregation of multiple semantic relationships can simplify the combination of semantic relationships, and supports to find the final reduced semantic relationships. This will reduce the problem-solving sample space, and speed up the work of the engine.

Another situation of semantic aggregation is to aggregate ontological items which are linked by logic connectors associated with some semantic relationships. The objective of aggregating ontological items is to reduce items, and generate the resulting ontological items.

Transformation from ontological item to another in one or many domains could be a mapping from an arbitrary keyword in the business ontological domain to its relevant items in the problem-solving domain. The basic idea for transformation of ontological items is as follows: given an input item, checking candidate ontological items by semantic relationships, and finding the suitable candidate as the output item.

The rules for the three types of transformations are discussed in [23].

5 Concluding Remarks

This paper told a story how agents and data mining can enhance each other, and how these two cutting edge technologies can be integrated through an ontological engineering point of view.

Integrating agents into data mining systems, the flexibility of data mining systems can be greatly improved. New data mining techniques can add to the systems dynamically, while the out-of-date ones can also be deleted from systems at run-time. It is a real plug-and-play. Equipping agents with data mining capabilities, the agents are much smarter and more adaptable. In this way, the performance of these agent systems can be improved.

This paper is a brief summary of what we have done in the practice of combining agents and data mining techniques. Obviously, more efforts are required to explore this new trend of research.

Acknowledgement

This paper is partially supported by the Ministry of Education of China Key Project 104160 and Deakin University CRGS grant.

References

1. M. Klusch, S. Lodi, and G. Moro, The Role of Agents in Distributed Data Mining: Issues and Benefits, *Proceedings of the IEEE/WIC International Conference on Intelligent Agent Technology*, IEEE CS Press, 2003, 211-217.
2. A. Ouali, Z. Ramdane-Cherif, A. Ramdane-Cherif, N. Levy, and M. Krebs, Agent Paradigm in Clinical Large-Scale Data Mining Environment, *Proceedings of the 2nd IEEE International Conference on Cognitive Informatics*, IEEE CS Press, 2003, 143-150.
3. M. Klusch, S. Lodi, and G. Moro, Agent-Based Distributed Data Mining: The KDEC Scheme, in: M. Klusch, S. Bergamaschi, P. Edwards, P. Petta (Eds.), *Intelligent Information Agents: The AgentLink Perspective, LNCS 2586*, Springer, 2003, 104-122.
4. Z. Zhang and C. Zhang, Constructing Hybrid Intelligent Systems for Data Mining from Agent Perspectives, in: N. Zhong and J. Liu (Eds.), *Intelligent Technologies for Information Analysis*, Springer, 2004, 327-353.
5. Z. Zhang, C. Zhang, and S. Zhang, An Agent-Based Hybrid Framework for Database Mining, *Applied Artificial Intelligence*, Vol. 17, No. 5-6, 2003, 383-398.
6. K. Ong, Z. Zhang, et al., Agents and Stream Data Mining: A New Perspective, *IEEE Intelligent Systems*, IEEE Press, 2005 (forthcoming).

7. Z. Zhang and C. Zhang, Agent-Based Portfolio Selection with Data Mining Ability, *Proceedings of 8th International Conference on Neural Information Processing*, Shanghai, China, 2001, 553-558.
8. N. R. Jennings, K. Sycara, and M. Wooldridge, A Roadmap of Agent Research and Development, *Autonomous Agents and Multi-Agent Systems*, Vol. 1, 1998, 7-38.
9. M. Luck, P. Mcburney and C. Preist, A Manifesto for Agent Technology: Towards Next Generation Computing, *Autonomous Agents and Multi-Agent Systems*, Vol. 9, 2004, 203-252.
10. W. Frawley, G. Piatetsky-Shapiro and C. Matheus, Knowledge Discovery in Databases: An Overview, *AI Magazine*, Fall 1992, 213-228.
11. M.-S. Chen, J. Han, and P. S. Yu. Data mining: an overview from a database perspective, *IEEE Trans. On Knowledge And Data Engineering*, Vol. 8, 1996, 866-883.
12. M. H. Dunham, *Data Mining-Introductory and Advanced Topics*, Prentice Hall, NJ, 2003.
13. U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, From Data Mining to Knowledge Discovery: An Overview, in: U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy (Eds.), *Advances in Knowledge Discovery and Data Mining*, MIT Press, Cambridge, MA, 1996, 1-34.
14. S. Dzeroski, Data Mining in a Nutshell, in: S. Dzeroski and N. Lavrac (Eds.), *Relational Data Mining*, Springer, 2001, 3-27.
15. I. Witten and E. Frank, *Data Mining: Practical machine learning Tools and Techniques with Java Implementations*, Morgan Kaufmann publishers, 2000.
16. H. Kargupta, B. Stafford, and I. Hamzaoglu, Web Based Parallel/Distributed Medical Data Mining Using Software Agents, *Proceedings of 1997 Fall Symposium*, American Informatics Association, 1997, *http://www.eecs.wsu.edu/~hillol/pubs.html*.
17. H. Kargupta, I. Hamzaoglu, and B. Stafford, Scalable, Distributed Data Mining Using an Agent Based Architecture, *Proceedings of Knowledge Discovery and Data Mining*, AAAI Press, 1997, 211-214.
18. A. Prodromidis, P. Chan, and S. Stolfo, Meta-learning in Distributed Data Mining Systems: Issues and Approaches, in: H. Kargupta and P.Chan (Eds.), *Advances in Distributed and Parallel Knowledge Discovery*, AAAI/MIT Press, 1999.
19. S. Bailey, R. Grossman, H. Sivakumar, and A. Turinsky, Papyrus: A System for Data Mining over Local and Wide Area Clusters and Super-Clusters, *Proc. International Conference on Supercomputing*, ACM Press, page 63, 1999.
20. Z. Zhang and C. Zhang, *Agent-Based Hybrid Intelligent Systems: An Agent-Based Framework for Complex Problem Solving*, LNAI 2938, Springer, 2004.
21. L. Cao, *Agent Service-Oriented Analysis and Design*, PhD Thesis, University of Technology, Sydney, Australia, 2005.
22. L. Cao, J. Ni, J. Wang, and C. Zhang, Agent-Services-Driven Plug-and-Play in F-TRADE, *Proceedings of 17th Australian Joint Conference on Artificial Intelligence, LNAI 3339*, Springer, 2004, 917-922.
23. L. Cao, D. Luo, C. Luo, and L. Liu, Ontology Transformation in Multiple Domains, *Proceedings of 17th Australian Joint Conference on Artificial Intelligence, LNAI 3339*, Springer, 2004, 985-990.