Proceedings of the 2008 IEEE
International Conference on Information and Automation
June 20 -23, 2008, Zhangjiajie, China

Multi-Agent based Automatic Evaluation System for Classification Algorithms

Fuzhen Zhuang[1,2], Qing He[1], Zhongzhi Shi[1]
[1]*The Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences*
[2]*Graduate University of Chinese Academy of Sciences*
{*zhuangfz,heq,shizz*}*@ics.ict.ac.cn*

***Abstract*— The current researches on classification usually focus on proposing novel algorithms and improving existent ones with better performances. However, although a classification algorithm is able to perform well for some given data sets, does it mean to any other data sets? And given several algorithm candidates, which one is the best for your classification problem? A practical solution to the above questions is to evaluate possible situations, which is extremely time-consuming and resource-consuming. In this paper we propose a distributed computing environment based on Multi-Agent technology to facilitate this evaluation process. In this computing environment we compare the performances of the same algorithm on different data sets, and different algorithms on the same data set. Experiments show that autonomic agents can run simultaneously and automatically on different computing hosts to achieve high availability, and this scheme can save the total evaluation time greatly. Therefore, this scheme will help us easily select the proper algorithm for a given classification problem according to different evaluation measures.**

***Index Terms*— Multi-Agent Technology; Distributed Environment; Performance Evaluation; Simultaneously and Automatically.**

I. Introduction

Formal and recent Distributed Data Mining (DDM), has attracted a lot of attentions among the data mining community [1], [2]. DDM refers to the mining of inherently distributed data sets, aiming to generate global patterns from the union set of locally distributed data. However, the security issue among different local datasets and the huge communication cost in data migration prevent moving all the datasets to a public site. Thus, the algorithms of DDM often adopt a computing paradigm of local processing and global synthesizing, which means that the mining process takes place at a local level and then at a global level where local data mining results are combined to gain global findings. Furthermore, the local processing often concerns multiple phases of data mining, including preprocessing, training and evaluation.

Since the development of Multi-Agent systems in the 80's, they have been considered as "societies of agents", i.e. as a set of agents that interact together to coordinate their behavior and often cooperate to achieve some collective goal. In nowadays, Multi-Agent technology has been more and more important for the application such as developing software applications and data mining, etc. Multi-Agent is used more and more widely in DDM environment, such as paper [3]. In that paper, a novel two-phase scheduling framework, based on the two-level architecture of an InterGrid, is presented for the Internet-wide distributed data mining, which shares the computing paradigm of local processing and global synthesizing. The DM IntraGrid with this internal scheduling algorithm has been implemented in a Multi-Agent system environment. In paper [4], we propose Multi-Agents based technology to realize the combination of Hyper Surface Classifiers (HSC) which is proposed in [5] and has many good properties [11]. Because of the intelligence of agents, in [6] Multi-Agent Simulation as a Tool for Modeling Societies is proposed.

It is well known that, the evaluation of mining algorithms is also very important in DM. Given a specifically mining task, it is hard to choose a suitable or better algorithm candidate. So how to select the best algorithm for the given mining task? The simple method is to evaluate possible algorithms serially, and then select the algorithm candidate according to some certain measurement. But they always have low efficiency and take too much time. In this paper, a parallel scheme using Multi-Agent technology for algorithm performance evaluation is proposed. Our scheme has many benefits compared to serial process, such as 1)it can save much time; 2)the algorithms can run simultaneously and automatically to achieve high availability; 3)it can dramatically reduce the interference of human beings; 4)we can receive the most directly and intuitionistic chart of evaluation results. Data mining process includes classification, clustering and etc. Our work focuses on the performance evaluation of classification algorithms.

This paper is organized as follows. In Section 2, we give an overview of our Multi-Agent System, including its architecture and some techniques employed. In Section 3, the implementations of the agents and scheme are presented in details. In Section 4, we give some experiments for performance comparison of mining algorithms using Multi-Agent technology, followed by our conclusions in Section 5.

978-1-4244-2184-8/08/$25.00 © 2008 IEEE.

II. An Overview of Multi-Agent System

Multi-Agent System (MAS) is a natural distributed computing environment, allowing agents running on different computing hosts simultaneously to achieve high availability. Moreover, MAS middleware helps to form a unified computing interface and makes the distributed applications more easily built. MAS is an open system, the structure of the system itself is capable of dynamically changing. Its components may not be known in advance, could change over time, and may be highly heterogeneous in that they are implemented by different people, at different times, using different software tools and techniques. In training and testing complex process, the output of an early job is the input of the following jobs. Thus, distributed execution of this process involves complex process control and management. To address this problem, a number of agents, specializing in solving constituent sub-problems in autonomous ways, need to be developed. They are then coordinated through standardized Agent Communication Language, which is well supported in a MAS environment by some communication protocols.

We adopted MAGE [7] as toolkit and MAS environment. MAGE is a middleware that facilitates the development of MA systems. Each instance of the MAGE runtime environment for each computing host is called a container, as it can contain several agents. Each container on that host presents the corresponding computing resource. Several containers can connect with each other through networks to form a natural distributed computing environment. From the view of programmers, it seems that all the containers are running on the same computer. We encapsulate each algorithm into the behavior of different agents. Such an agent can be regarded as an algorithm entity that can run on any host. To initiate an agent on a host means to perform the behavior of that agent, and thus utilize the corresponding computing resource. In paper [8], we implement this system in an established Multi-Agent system environment, in which the reuse of existing data mining algorithms is achieved by encapsulating them into agents.

III. Systemic Framework and Scheme Design

A. Systemic Framework

With the Agents designed in our scheme, there are several types of agents, including ScheduleAgent, LoadMonitorAgent, ExecuteAgent, SenderAgent, EvaluateAgent and other dynamically generated agents such as various mining algorithm agents. How do these agents coordinate their behavior and often cooperate with each other? The whole systemic framework chart is shown in Figure.1.

The task repository describes all the task configuration information built for the system. What the configuration files describe includes the training data file and testing data file information, various mining algorithms parameters and other task description. In the distributed environment, every computing host is an autonomic entity and can be regarded as a container in the Multi-Agent System. A container contains several inherent agents, and also can create new agent dynamically to do mining task. When the task has been done, the corresponding agent will be killed automatically. The sketch of container with inherent agents and dynamic agents is in Figure.2. A container includes three inherent agents, ScheduleAgent, LoadMonitorAgent and ExecuteAgent, this means that every computing host in the distributed environment are at equal viscounty. There will dynamically and automatically create corresponding algorithm agents when they execute mining task. Because of the equipollence of all hosts in the distributed environment, any one of the hosts can be selected as a monitor host when we start the Multi-Agent System.

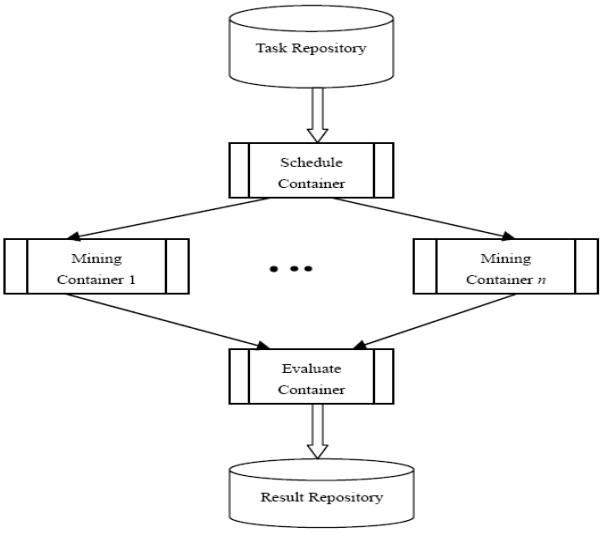

Fig. 1. The holistic framework chart.

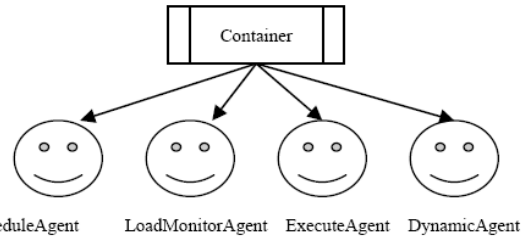

Fig. 2. The sketch of container with agents.

B. The Total Design of Scheme

There are several kinds of agents mentioned above, and what roles do they play in our system? When the system

starts, one of the computing hosts acts as a monitor host. Firstly it loads the configuration information to its cache or RAM, the magnitude of all task are defined by the way mentioned in [3], [8]; Secondly, the LoadMonitorAgents control and inspect the status of task executing on every hosts in the distributed system, such as the real-time workloads of every host and etc. The LoadMonitorAgent in the monitor host (MLoadMonitorAgent) sends message to other LoadMonitorAgents in the distributed environment, inquiring the workloads of every slavery host. Then the LoadMonitorAgents in the mining containers reply workloads to the MLoadMonitorAgent. Thirdly, the ScheduleAgent in the monitor host (MScheduleAgent) starts to work, and it schedules the task for the slavery hosts in the distributed system according to the feedback information from the LoadMonitorAgents. Then the SenderAgent starts to send the corresponding task to slavery hosts. When the task reaches the host, if the workload of this host is zero, then the ExecuteAgent starts corresponding algorithm agent to execute the task; else the task is put into the task queue and waits for the host free. When the system starts, all the agents run autonomically, parallelly and automatically.

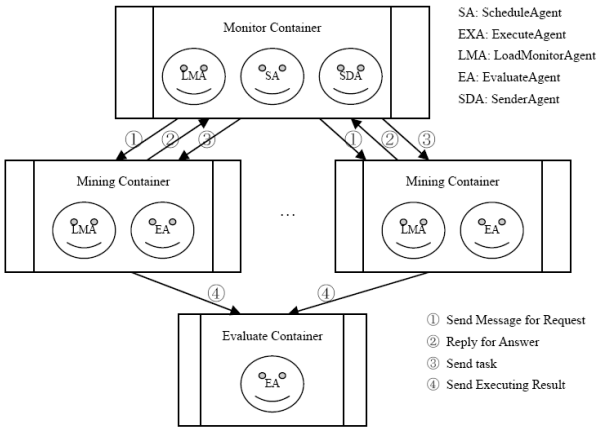

Fig. 3. Intuitionistic Chart for Running Process of Agents.

After all the tasks are scheduled, they are executed by the algorithm agents in the corresponding host. Sometimes, the tasks are not scheduled equably, so we have to reschedule the task. In this case, in our system the rescheduling scheme is executed which is similar to the previous phase. The host which has the heaviest workload acts the monitor host, and then reschedules the task to other free host. The detail of scheduling scheme and how to balance the workload in every host can be found in [3], [8]. EvaluateAgent monitors the running status of all mining agents. When all the mining tasks are finished, the EvaluateAgent summarize and evaluate the results, then output the results and the visualized chart. In the whole running processes, different agents in the Multi-Agent System communicate with each other according to the standard agent communication language (SACL). Because of the use of Multi-Agent system environment, all the containers are seemed to run on the same computer, programmers can conveniently control all hosts in the distributed environment. On the other hand, autonomic agents can run simultaneously on different computing hosts to achieve high availability. Moreover, it can improve the efficiency and dramatically reduce the total executing time.

To illuminate the running process of agents, a more intuitionistic chart is shown in Figure.3. The ExecuteAgent can dynamically create and kill algorithm agents, so the algorithm agents and unnecessary agents are not shown in the chart. Some kinds of algorithm agents are shown in Figure.4.

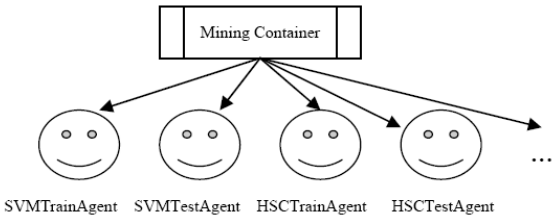

Fig. 4. Several Kinds of Algorithm Agents.

IV. Experiments

The first three experimental data sets are provided by the UCI repository data sets, and four experiments have been done in this paper. In our experiment, there are only six mining algorithm agents; because of the well scalability, you can add any other mining algorithm agents to the system easily and conveniently. We also evaluate the consumed time taken by serial algorithm and our scheme in the experiments.

In the first experiment, the breast-cancer data set is selected, which has 683 samples. We deliberately select 72 samples which are supported vectors obtained by SVM^{lib} [12] on the parameters (radial basis-function, $c = 1.0, \gamma = 0.125$) for training and the rest 611(683-72) for testing. The results of accuracy including close test and open test are shown in Table.I. The histogram is also shown in Figure.5.

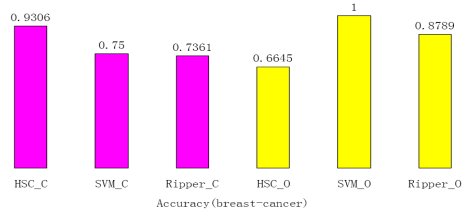

Fig. 5. Intuitionistic Histogram of Accuracy for Breast Data Set.

In the figure, where the postfix "C" stand for "Close test", i.e., Training, and the postfix "O" stand for "Open test", i.e., Testing; for example HSC_C stand for "HSC close test" and HSC_O stand for "HSC open test". In the following context we will adopt the same naming convention if there is not particular specification. For the first data set, SVM performs a perfect generalization, but HSC and Ripper perform not so well.

To give more profound impression for our system, Figure.6 and Figure.7 show the visualization interface when the system runs.

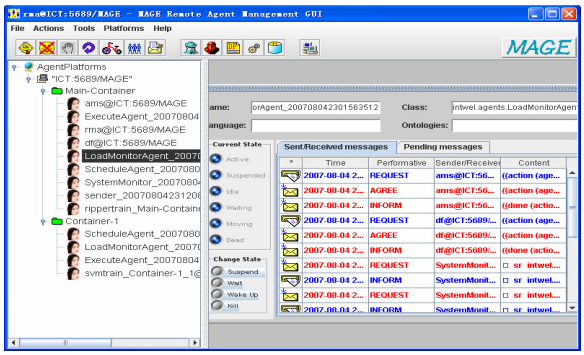

Fig. 6. The visualization interface of executing process.

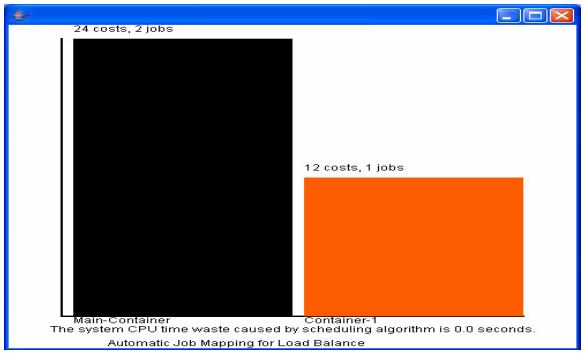

Fig. 7. The visualization interface of executing process.

In the second experiment, the waveform data set is selected, which has 5000 samples. 4522 samples, which is Minimal Consistent Subset for a disjoint Cover set (MCSC) proposed in [9], [10], are selected for training and the rest 478 for testing. The data set is preprocessed from high dimensional data to three dimensions by the method in [11]. The results of accuracy including close test and open test are shown in Table.I. For the second data set, HSC performs a perfect generalization, but SVM^{lib} and Ripper perform not so well. The visualized chart produced by our system of close and open test are shown in Figure.8.

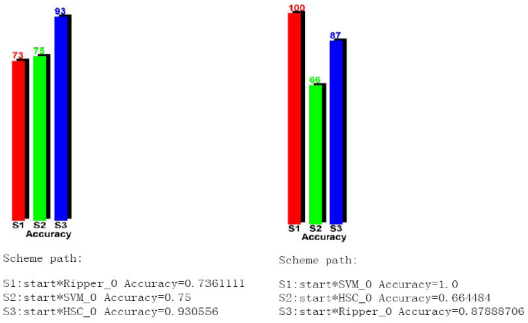

Fig. 8. Intuitionistic Histogram of Results for Letter-Recognition Data Set.

In the Third experiment, the sub data set of Letter-Recognition is selected, which total size is 20000. 6000 samples are randomly selected from the original data sets for our experiment; and the selected sub data set are also randomly split. The results of accuracy including close test and open test are shown in Table.I. Because of the randomly selected data set, the generalizations of three algorithms are not as regular as the anterior two experiments; but it seems SVM performs better.

TABLE I
ACCURACY OF ALL DATA SETS TESTED BY HSC, SVM, RIPPER

data set		HSC	SVM	Ripper
Break-Cancer	Training	93.06%	75%	73.61%
	Testing	66.45%	100%	87.89%
Waveform	Training	93.34%	47.35%	62.98%
	Testing	99.37%	62.34%	80.75%
Letter-Recognition	Training	63.78%	99.86%	85.52%
	Testing	50.74%	92.9%	76.9%
Ten-Spirals	Training	100%	10.03%	97.4%
	Testing	100%	9.96%	97.32%

In the fourth experiment, the data set is constructed in ten-spirals described in paper [11]. The data set has three dimensions and the size of training and testing data sets are 33750 and 16880, respectively. The ten spirals $K_i(1 \leq i \leq 10)$ in coordinates are:

$$\begin{aligned} K_i &: \rho = \theta + (i+1)\pi/5 \\ z &= \rho, \frac{\pi}{2} \leq \rho \leq 8\pi, (1 \leq i \leq 10). \end{aligned} \tag{1}$$

And the sketch map of samples generated by ten spirals is shown in Figure.9. The results of accuracy including close test and open test are shown in Table.I. We can find that the data set is very suitable for HSC and Ripper, but

unfortunately, it is not so good for SVM.

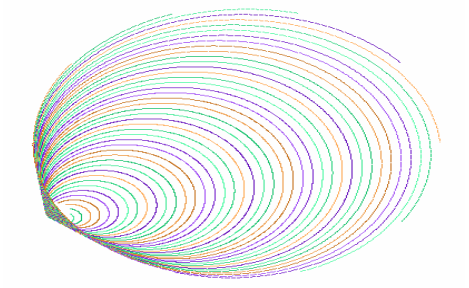

Fig. 9. The sketch map of samples generated by ten spirals.

Why SVM algorithm is so poor performance on ten-spirals data set, we analyze the characteristic of ten-spirals data set. For spiral distribution and overlapping of the data, which can not be classified by linear classifiers and too complex for SVM algorithm. To classify the data, SVM has to map the data from low dimensional space to very high dimensional space, which lead to the complexity and the poor performance of SVM. But for HSC and Ripper, they are all covering algorithms and don't need the data linearly classified. Experiments in [11] also confirm that HSC can performance well on such data sets. Although we can improve the accuracy of SVM algorithm by changing the parameters, the expends of executing time are stupendous. In the experiment, we use the parameters ($c = 1000, \gamma = 0.125$) instead of ($c = 1.0, \gamma = 0.125$), focusing on the wrong predicted samples. Though the accuracy of close test and open test are up to 52.59% and 52.59% respectively, but the executing time drastically rises to more than 2.0×10^4 seconds; compared to the corresponding time consumed by HSC and Ripper in Table.III, it is more than one hundred times with respect to HSC algorithm. Therefore the ten-spirals data are more appropriate for HSC and Ripper, rather than SVM. Table.II describe the performance of the same algorithm on different data sets. The intuitionistic histograms are shown in Figure.10.

TABLE II

ACCURACY OF ALGORITHMS ON DIFFERENT DATA SETS

	Breast-Cancer	Waveform	Letter-Recognition	Ten-Spirals
HSC_C	93.06%	99.34%	63.78%	100%
HSC_O	66.45%	99.37%	50.74%	100%
SVM_C	75%	47.35%	99.86%	10.03%
SVM_O	100%	62.34%	92.9%	9.96%
Ripper_C	73.61%	62.98%	85.52%	97.4%
Ripper_O	87.89%	80.79%	76.9%	97.32%

From the results of the four experiments, we can find that it is easy to implement performance evaluation for classification algorithms using Multi-Agent Technology. Also we can find that different data sets, the algorithms perform observably different. For the breast-cancer and letter-recognition data sets, SVM outperforms HSC and Ripper algorithms. But for the waveform and ten-spirals data sets, HSC algorithm outperforms SVM; in a whole, Ripper algorithm performs in the middle level on the data sets selected in our experiments. Experiments testify that the selection of sample set intensely influences the mining algorithm performance; and the selection of proper algorithm is also very important when executing a specifically data mining task.

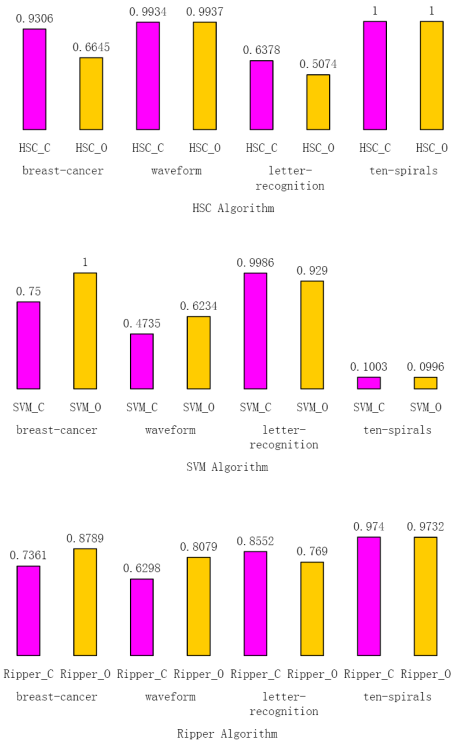

Fig. 10. Intuitionistic Histograms of HSC, SVM and Ripper on different data sets.

To show the advantages of our scheme, Table.III describes the executing time of two methods, including serial computing and our scheme (two containers are started in our system). The results show that our scheme can efficiently reduce the executing time, and the total consumed time is constrained by the worst executing time of single algorithm. In the ideal case, the total consumed time is nearly equal to the worst executing time of single algorithm and it can save the executing time of other algorithms. From the results, we can find that each algorithm sometimes performances poor for some data set. So improving the performance of each algorithm by sampling is our further researches.

TABLE III

THE CONSUMED TIME OF SERIAL COMPUTING AND OUR SYSTEM, AND SAVED TIME BY OUR SYSTEM

data set		Serial Computing(HSC +SVM+Ripper)(seconds)	Our System (seconds)	Saved Time (seconds)
Break-Cancer	close test	6.328 (2.953+1.797+1.578)	5.188	1.14
	open test	17.469 (4.672+7.422+5.375)	11.141	6.328
Waveform	close test	667.765 (320.86+328.765+18.14)	340.359	327.405
	open test	345.337 (311.86+21.282+12.235)	325.703	19.674
Letter-Recognition	close test	407.218 (114.953+201.453+90.812)	209.281	197.937
	open test	247.953 (113.578+81.031+53.344)	169.5	78.453
Ten-Spirals	close test	3418.08 (158.016+1679.06+1581)	1747.19	1670.89
	open test	2943.95 (195.937+1431.17+1316.84)	1519.68	1424.27

V. CONCLUSION

The Multi-Agent technology is used more and more widely in distributed data mining. In this paper we propose a scheme for evaluation of classification algorithms using Multi-Agent technology. Given a particular data set, and if there are several algorithm candidates, by this evaluation scheme you can efficiently select the proper classification algorithm for this classification task. Experiments show that the autonomic agents can run simultaneously and automatically on different computing hosts to achieve high availability, and it can save the total executing time greatly, and dramatically reduce the interference of human beings.

ACKNOWLEDGMENT

This work is supported by the National Science Foundation of China (No. 60435010, 60675010), the 863 Project (No.2006AA01Z128), National Basic Research Priorities Programme (No. 2007CB311004) and the Nature Science Foundation of Beijing (No. 4052025).

REFERENCES

[1] Y.Fu, "Distributed data mining: An overview". IEEE TCDP newsletter(2001).

[2] M.Cannataro, A.Congiusta, A.Pugliese, D.Talia, P.Trunfio, "Distributed data mining on grids: Services, tools, and applications", IEEE Transactions on Systems, Man and Cybernetics 34 (6) (2004) 2451-2465.

[3] P.Luo, K.Lu, Z.Z.Shi, Q.He, "Distributed data mining in grid computing environments", Future Generation Computer Systems 23 (2007) 84-91.

[4] Q.He, X.R.Zhao, P.Luo, and Z.Z.Shi, "Combination Methodologies of Multi-Agent Hyper Surface Classifiers", Design and Implementation Issues, V. Gorodetsky et al. (Eds.): AIS-ADM 2007, LNAI 4476 (2007)100-113.

[5] Q.He, Z.Z.Shi, L.A.Ren, "The classification method based on hyper surface", IJCNN '02. Proceedings of the 2002 International Joint Conference on Neural Networks, (2002) 1499-1503.

[6] Alex.Drogoul, J.Ferber, "Multi-agent simulation as a tool for modeling societies: Application to social differentiation in ant colonies". In Actes du Workshop MAAMAW'92, 1992.

[7] Z.Z.Shi, H.Zhang, Y.Cheng, Y.Jiang, Q.Sheng, and Z.Zhao, "MAGE: An Agent-Oriented Programming Environment". In Proceedings of the IEEE International Conference on Cognitive Informatics, New York (2004) 250-257.

[8] P.Luo, K.Lu, R.Huang, Q.He, and Z.Z.Shi, "A Heterogeneous Computing System for Data Mining workflows in Multi-Agent Environments". Expert Systems 23 (2006) 258-272.

[9] Q.He, X.R.Zhao, Z.Z.Shi, "Sampling Based on Minimal Consistent Subset for Hyper Surface Classification", International Conference on Machine Learning and Cybernetics, pp.12-17, 2007.

[10] Q.He, F.Z.Zhuang, X.R.Zhao, Z. Z.Shi, "Enhanced Algorithm Performance for Classification Based on Hyper Surface Using Bagging and AdaBoost". International Conference on Machine Learning and Cybernetics, pp.3641-3646, 2007.

[11] Q.He, Z.Z.Shi, L.A.Ren, E.S.Lee, "A Novel Classification Method Based on Hyper Surface". Int. J. of Mathematical and Computer Modeling 38 (2003) 395-407.

[12] The SVM^{lib} is available at http://www.csie.ntu.edu.tw/ cjlin/libsvm/.