Minority Game Data Mining for Stock Market Predictions

Ying Ma, Guanyi Li, Yingsai Dong and Zengchang Qin*

Intelligent Computing and Machine Learning Lab
School of Automation Science and Electrical Engineering
Beihang University, Beijing, 100191, China
* zcqin@buaa.edu.cn

Abstract. The Minority Game (MG) is a simple model for understanding collective behavior of agents in an idealized situation for a finite resource. It has been regarded as an interesting complex dynamical disordered system from a statistical mechanics point of view. In previous work, we have investigated the problem of learning the agent behaviors in the minority game by assuming the existence of one "intelligent agent" who can learn from other agent behaviors. In this paper, we propose a framework, Minority Game Data Mining (MGDM), that assumes the collective data are generated from combining the behaviors of variant groups of agents following the minority game. We then apply this framework to real-world time-series data analysis by testing on a few stocks from the Chinese market and the US Dollar-RMB exchange rate. The experimental results suggest that the winning rate of the new model is statistically better than a random walk.

1 Introduction

An economic market is regarded as a complex adaptive system (CAS) by the physicists and computer scientists from the econophysics [10] community. It is an interdisciplinary research field that applies theories and methods originally developed by physicists in order to solve problems in economics, usually those including uncertainty or stochastic processes and nonlinear dynamics. Agent-based models of complex adaptive systems provide invaluable insight into the highly non-trivial collective behavior of a population of competing agents, this has been used in experimental economics [15, 5] financial market modeling [8, 9] and market mechanism designs [11].

Agent-based experimental games have attracted much attention among scientists in different research areas [10], e.g., in financial modeling people are trying to solve the problem of analyzing the real market system where involving agents with similar capability are competing for a limited resource. One of the most important issues is: in such a complex system, every agent knows the history data in the market and must decide how to trade based on this global information. Among these agent-based models, minority game (MG) [4] is an important model in which an odd number N of agents successively compete to be in the

minority. It can be regarded as a simplified version of EI Farol bar problem [1], in which a number of people decide weekly whether go to the EI Farol bar to enjoy live music in the risk of staying in a crowd place or stay at home. As a new tool for learning complex adaptive systems, the minority game has been applied to variety areas [8, 9] especially in financial market modeling [2, 3]. But this model has limitations that agents are always assumed to be identical and it is hardly used in prediction.

In order to design a useful prediction tool by using MG, the first thing we notice is that, it is unrealistic to assume all agents use the same patterns as their strategies. In real-life scenarios, some agents make random decisions and some groups employ similar strategies. The complexity of marketing world is embodied in existence of varieties types of agents using strategies based on their own rules. Also it is unrealistic to have the whole population following the same rules of a game. There could be some more intelligent ones who are standing out by learning from the others.

In this paper, we propose a model that efficiently figures out limitations of previous models when apply to the real markets. Through observing the dynamics, we increase complexity of this model by divided the real market into several diverse types of agents. Using machine learning and data mining, an intelligent agents can analyze and estimate the real dynamic markets to maximize own profits in terms of winning probability. It is a new way to understand the relationship of micro-behaviors and macro-behaviors by utilizing minority game model and machine learning. This paper is organized as the following: section 2 introduces the minority game and we propose learning methods of using decision tree and genetic algorithm for discovering the composition of agents in order to predict the macro-behavior of the MG. In section 3, a series of experimental results are presented using real-world data from stock price and currency exchange rate. We verify the effectiveness of this learning mechanism and conclusions are given in the end.

2 Learning in the Minority Games

EI Farol bar problem proposed by Author [1] is one of the best-known experimental game models in economics, in which a number of people decide weekly whether go to the EI Farol bar to enjoy live music in the risk of staying in a crowd place or stay at home. Formally: N people decide independently to go or stay each week, they have two actions: go if they expect the attendance to be less than an integer $[\alpha N]$ (where $0 < \alpha < 1$) people or stay at home if they expect it will be overcrowded. There is no collusion or prior communication among the agents; the only information available is the numbers who came in past weeks. Note that there is no deductively rational solution to this problem, since given only the numbers attending in the recent past.

One simplified version of the El Farol Bar problem is the Minority Game proposed by Zhang and Challet [4]. In the minority game, there is an odd number of players and each must choose one of two choices independently at each

turn. The players who end up on the minority side win. The minority game examines the characteristic of the game that no single deterministic strategy may be adopted by all participants in equilibrium [16]. In this section, we will set an environment that is populated by a few diverse groups of agents. We aim to design one type of intelligent agents that can take advantage from real stock markets by learning from the patterns of other agents and make choices in order to be on the minority side. In the next section, the mathematical treatments of the MG will be introduced in details.

2.1 Strategies of Agents

A population of an odd number of N agents decide between two possible options, say to attend room A or B at each round of the game. Formally, at the round t ($t = 1, \ldots, T$): each agent need to take an action $a_i(t)$ for $i = 1, \cdots, N$, to choose room A or B.

$$a_i(t) = \begin{cases} A & \text{Agent } i \text{ choose room } A \\ B & \text{Agent } i \text{ choose room } B \end{cases} \tag{1}$$

In each round of the game, the agents belonging to the minority group win. The winning outcomes can be considered as a binary function $w(t)$. Without loss of generality (w.l.o.g), if A is the minority side, we define that the winning outcome is 0, otherwise, it is 1. In this paper, the winning outcomes are known to public.

$$w(t) = \begin{cases} 0 & \#(a_i(t) = A)|_{i=1,\ldots,N} \leq (N-1)/2 \\ 1 & \text{otherwise} \end{cases} \tag{2}$$

where $\#()$ is the function for counting numbers: for all the agents (i runs from 1 to N), if the number of agents that satisfy the condition $a_i(t) = A$ is less than or equal to $(N-1)/2$, then $w(t) = 0$; otherwise, it is $w(t) = 1$. We assume that agents make choices based on the most recent m winning outcomes $h(t)$, which is called *memory* and m is the *length of memory*.

$$h(t) = [w(t-m), \ldots, w(t-2), w(t-1)] \tag{3}$$

Given the outcome $w(t)$ at the moment t, agent i keeps a record $r_i(t)$ that tells whether it has won the game or not.

$$r_i(t) = \begin{cases} T & \text{Agent } i \text{ wins at time } t \\ F & \text{Agent } i \text{ loses at time } t \end{cases} \tag{4}$$

In minority game, we usually assume that each agent's reaction based on the previous data is governed by a "strategy" [2–4]. Each strategy is based on the past m-bit memory which are described as a binary sequence. Every possible m-bit memory are mapped in correspond to a prediction of choosing room A or B in the next round. Therefore, there are 2^{2^m} possible strategies in the strategy space. Agents in the same fixed strategy group share one strategy randomly selected from the strategy space. Given the memory $h(t)$, the choice for the agent i guided by the strategy S is denoted by $S(h(t))$. Table 1 shows one possible strategy with

$m = 4$. For example, $h(t) = [0000]$ represents that if the agent who choose A in the latest four steps win, the next round (at round t) choice for this agent will be $S([0000]) = A$. A strategy can be regarded as a particular set of decisions on the permutations of previous winning outcomes.

Table 1. One possible strategy with $m = 4$: the current choice of agent is decided by its previous 4 step memory.

$h(t)$	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
$S_1(h(t))$	A	A	B	B	B	A	B	A	B	A	A	A	B	A	B	B

Giving the history data of all winning outcomes, training data for an agent can be sampled by a sliding window with size m on each time step. At t round, the target value (either A or B) is the agent's actual choice at the current round. Therefore, training set for agent i can be formally defined as:

$$\mathcal{D}_i = \{(h(t), a_i(t))\} \qquad for \quad t = 1, \ldots, T \tag{5}$$

Based on this training set, we hope to find an effective algorithm to learn other agents' strategies.

2.2 Decision Tree Learning

Decision tree learning is one of the simplest and most effective learning algorithms. It has been widely used in numerous machine applications for its simplicity and interpretability [13] (A decision tree can be decomposed into a set of rules). In this paper, the tree model we use is actually probability estimation tree [12] because we are considering the probability distribution over two possible choices of A and B.

Figure 1 shows a decision tree for modeling behaviors of agent i based on the training date . For each branch $W = [w(t-m), \ldots, w(t-2), w(t-1)]$, there is an associated probability distribution on possible agent's choices (i.e., A or B) that is calculated based on the proportion of these two kinds of data falling through the branch. Or formally:

$$P(A|W) = \left. \frac{\#(h(t) = W \wedge a_i(t) = A)}{\#(h(t) = W)} \right|_{(h(t), a_i(t)) \in \mathcal{D}_i} \tag{6}$$

where $\#()$ is the same counting function appeared in eq. 2, therefore, eq. 6 means that: given a branch, it looks for all matching string from the training data and checks how many of them chose A. If agent i follows a particular fixed strategy and game is repeated for n ($n \gg 0$) rounds. We then can get a fair good estimation of the behaviors of agent i given the 3-step memory training data.

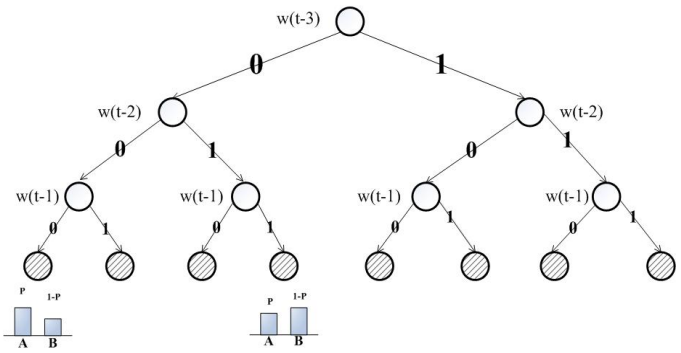


Fig. 1. A schematic illustration of a decision tree where each leaf gives the probability distribution over two classes.

For each agent i, its current choice $a_i(t)$ can be predicted by the decision tree learning based on the training data $\mathcal{D}_i$(see eq. 5). At t round of the game, the probability that the intelligent agent choose A, $P_I(A)$, is calculated based on its estimation of other agents' choices $a_i(t)$ where $i = 1, \ldots, N$.

$$P_I(A) = 1 - \left. \frac{\#(a_i(t) = A)}{\#(a_i(t) = A) + \#(a_i(t) = B)} \right|_{i=1,\ldots,N} \tag{7}$$

and $P_I(B) = 1 - P_I(A)$. The above equation can be interpreted that the intelligent agent will go to the room that most of agents won't go based on its predictions of other agents' behaviors. Simply, the intelligent agent makes choice based on its estimation $P_I(A)$ and $P_I(B)$.

$$a_I(t) = \begin{cases} A & P_I(A) > P_I(B) \\ B & \text{otherwise} \end{cases} \tag{8}$$

The accuracy of wining for the intelligent agent I can be obtained by:

$$AC_I(t) = \left. \frac{\#(r_I(t) = T)}{\#(r_I(t) = T) + \#(r_I(t) = F)} \right|_{t=1,\ldots,T} \tag{9}$$

This estimation is based on *complete information*, i.e., all records of agents' choices and history outcomes are known to public, Our previous work [7] has shown the experimental results in diverse environments. These results indicate intelligent agent can outperform other agents using this simple learning algorithm.

2.3 Genetic Algorithm for Minority Game Data Mining

In the previous section, we designed an intelligent agent that uses machine learning method to learn the patterns of other agents in the MG. However, the complete information is an ideal situation, in most cases, we can only obtain the

collective data $w(t)$. In this section, we propose a framework that assumes the macro-behavior can be decomposed into several small groups of agents employing the same strategy. A *Genetic Algorithm* [6] is used to estimate the parameters of this decomposition.

Genetic Algorithms (GA), developed by Holland [6], is a fast and intelligent way to search one suitable parameter in large space of parameters. We use a vector of parameters to represent the number of agents in each group and the strategy they use, a GA is used to optimize these parameters in order to find the most likely combinations of single behaviors that could generate a certain macro-level sequence. After getting the most likely compositions, intelligent agent can make decision by using decision trees.

In our model, intelligent agent only use the information of winning outcomes $w(t)$ and a guessed maximum number of groups using fixed strategies K, such that the agents can be divided into $K+1$ groups: $\{G_R, G_1, \ldots, G_K\}$, where group G_R is the group of random agents and G_k for $k = 1, \ldots, K$ employs the strategy S_k. A chromosome x is consisted by the the following parameters: percentage of random agents P_R (among all agents), corresponding percentage of agents with one certain fixed strategy P_{S_k} and the strategy S_k.

$$\mathsf{x} = \{P_R, P_{S_1}, S_1, \ldots, P_{S_K}, S_K\}$$

Figure 2 illustrates that, the collective data is a combination of choices from a diverse group of agents. Based on the give history sequence $w(t)$, intelligent agent use GA to explore all possible combinations of subgroups in order to find original compositions of markets. Intelligent agent then uses the information to make predictions and be in the minority side.

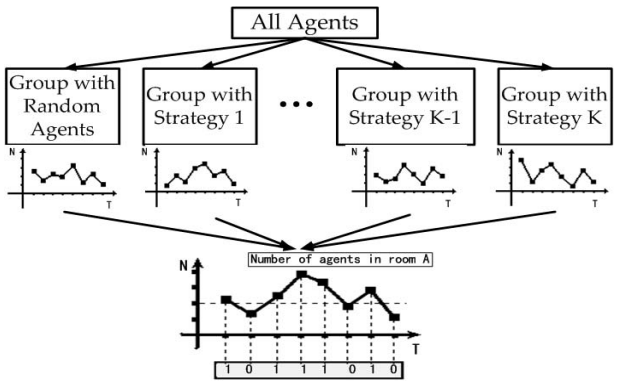


Fig. 2. The process of generating collective data. The whole system can be divided into $K + 1$ groups. At each round, the collective data is considered as an aggregation of actions of every group of agents. When collective data indicates A is the minority side, the sequence of winning outcomes will be 0, otherwise, it will be 1

The fitness function is calculated as the following: at t round of the game, in order to evaluate one chromosome $\mathbf{x}_j$ ($j = 1, \ldots, J$ where J is the population size in the GA), we run the MG with the parameter setting given by $\mathbf{x}_j$ to obtain the history of winning outcomes $y_j(t)$. The fitness function is defined based on the comparisons between $y_j(t)$ and the actual sequence of winning outcomes $w(t)$: for t runs from 1 to a specified T, once $y_j(t) = w(t)$, we add one point to $f(\mathbf{x}_j)$, formally:

$$f(\mathbf{x}_j(t)) \leftarrow \begin{cases} f(\mathbf{x}_j(t)) + 1 & \text{if: } y_j(t) = w(t) \\ f(\mathbf{x}_j(t)) & \text{otherwise} \end{cases} \tag{10}$$

At each round t, the best chromosome $\mathbf{x}_j^*$ is selected from the pool:

$$\mathbf{x}^*(t) = \arg\max_j f(\mathbf{x}_j(t)) \quad for \quad j = 1, \ldots, J$$

The best chromosome $\mathbf{x}^*(t)$ gives the parameters can be interpreted into a combination of the subgroups.

best possible complete information of a MG so that the intelligent agent can learn with decision trees discussed in section 2.

3 Experiments on Real Markets

This learning process points us a new way of using the minority game model and evolutionary optimization in understanding the relationship between micro-data and macro-data. The collective behaviors of MGs can be decomposed into several micro-behaviors from different group of agents. Combining the behaviors of these groups may generate complex and seemingly unpredictable macro-behaviors. Given a sequence of history wining outcomes, by using genetic algorithm, we can find the most likely combinations of single behaviors that could generate this sequence. Many real-world complex phenomena are related to the minority game [8, 9, 4] such as stock market and currency exchange rate. Although the macro-level data are seemingly random and unpredictable, we could build models with the MGDM framework to reconstruct the mechanism for generating these data and predict possible future results.

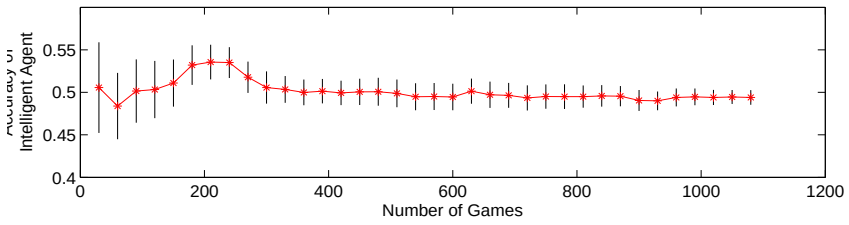


Fig. 3. Accuracy of intelligent agent with randomly generated sequence.

3.1 Experimental Results

In the following experiments, we randomly selected 12 stocks from the Chinese market and US Dollar-RMB (Chinese Renminbi) exchange rate to test the MGDM model. The stock prices are from a downloadable software [17] and the exchange rate from the website [18]. For each stocks (with stock index) or exchange rate, we test successive 1100 trading days from the beginning date listed in table 2.

Each round of the game represents one trading day. Only given macro-level data $w(t)$, intelligent agent use MGDM to predict the most likely winning choice next round. Suppose the opening price is V_b and the closing price is V_f. Every trading day t, the fluctuation of the stock or exchange rate can be transferred to $w(t)$ by:

$$w(t) = \begin{cases} 1 & V_f \geq V_b \\ 0 & otherwise \end{cases}$$

At the end of every trading day t, intelligent agent select the best chromosome $\mathbf{x}^*(t)$ to make prediction for the next day based on equation 10.

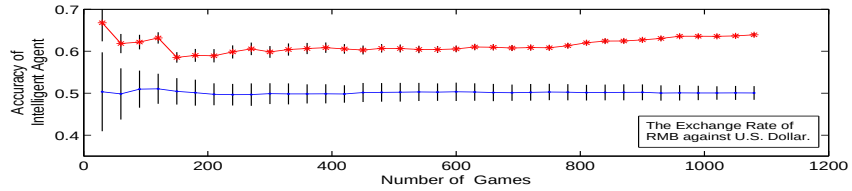
The parameters for MGDM are as the following: the guessed maximum number of groups using fixed strategies $K = 20$, strategies S_k generated with memory length $m = 3$ and the population size $J = 50$. In order to avoid the influence of randomness from GA, we run the whole experiments for 30 times and the mean (marked with stars) and the standard deviations are recorded and plotted in figure 4. The final results after 1100 trading days are also shown in table 2.

In order to test the statistical significance, we first tested on a random sequence and the mean accuracy associated with standard deviations of the intelligent agent is shown in figure 3. It is gradually approaching 50% - which is true, because no learning method could take any advantage from a totally random market. Agents must have learnt something useful as long as the accuracy is over 50%. In figure 4, all prediction results are compared with such random sequence (or random walk) to test effectiveness of the MGDM framework.

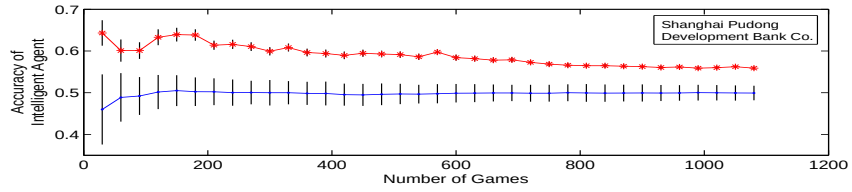
3.2 Data Analysis

Figure 4 shows all the test results using data from real markets where each experiments contains 1100 steps (trading days). In each graph of figure 4, the upper curve indicates the mean accuracy (associated with the range of plus or minus standard deviations) of the intelligent agents. The lower curve shows the accuracy when intelligent agent make random choice without using any learning methods. As we can see that, in all the graphs showed in figure 4, intelligent agent using MGDM models performs considerably better than a random walk (by a margin from 2% to 14%), detailed results are listed in table 2, demonstrating the superiority the model in highly unpredictable stock markets.

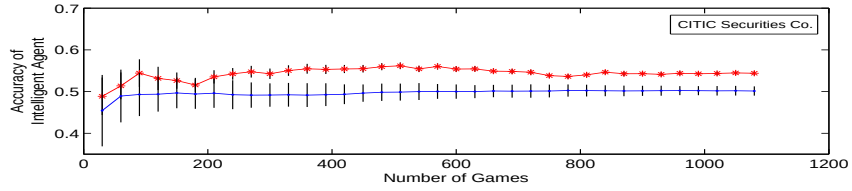
In the graph of the exchange rate of RMB against U.S. Dollar (subfigure 1), the mean accuracy is up to 63.93% along with the ascendant trend. This kind of markets may has a strong pattern inside. The mean accuracy of most of the graphs is around range from 53% to 55%, e.g., the mean accuracy of the


Accuracy of Intelligent Agent
Number of Games
The Exchange Rate of RMB against U.S. Dollar.

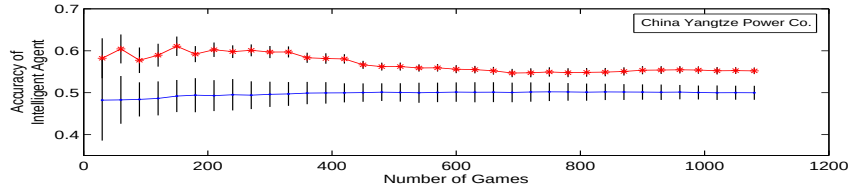
1


Accuracy of Intelligent Agent
Number of Games
Shanghai Pudong Development Bank Co.

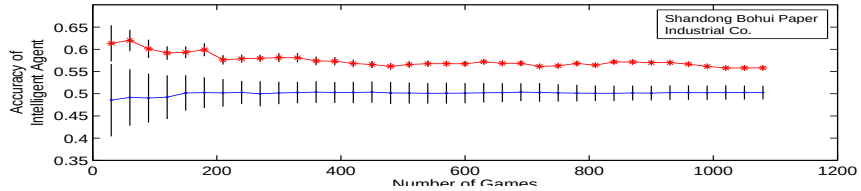
2


Accuracy of Intelligent Agent
Number of Games
CITIC Securities Co.

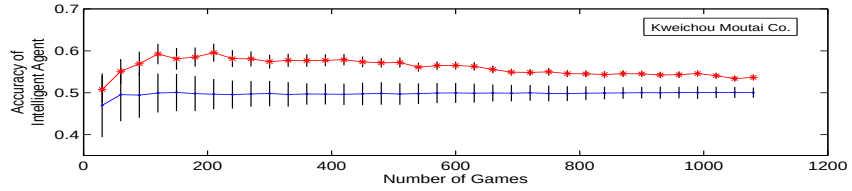
3


Accuracy of Intelligent Agent
Number of Games
China Yangtze Power Co.

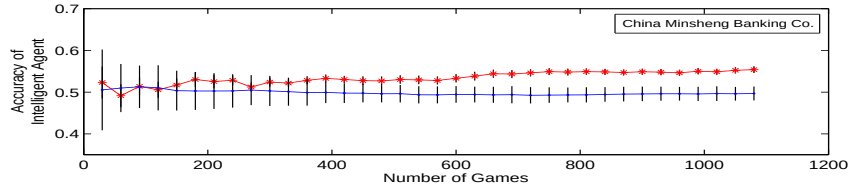
4


Accuracy of Intelligent Agent
Shandong Bohui Paper Industrial Co.

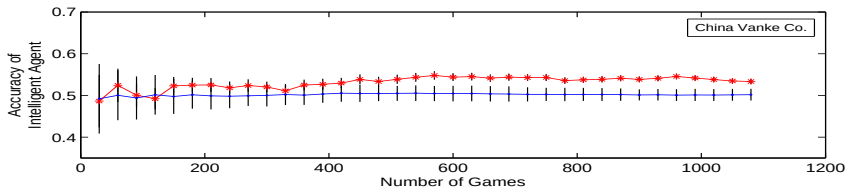
5


Accuracy of Intelligent Agent
Number of Games
Kweichou Moutai Co.

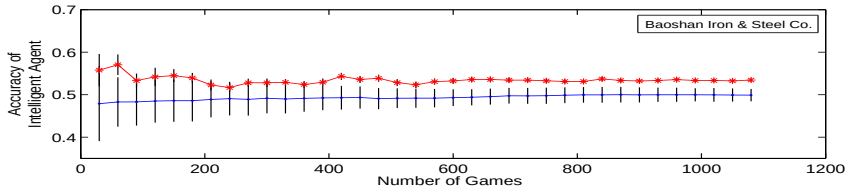
6


Accuracy of Intelligent Agent
Number of Games
China Minsheng Banking Co.

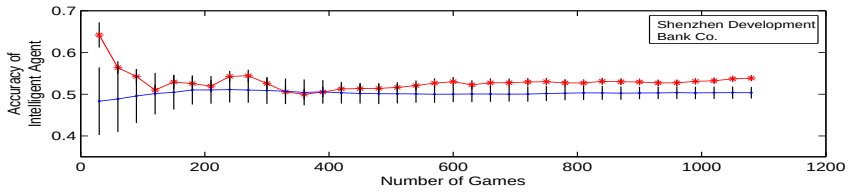
7



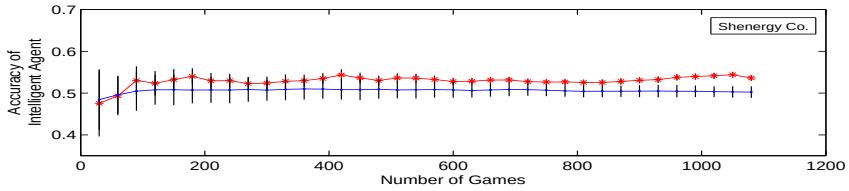
8



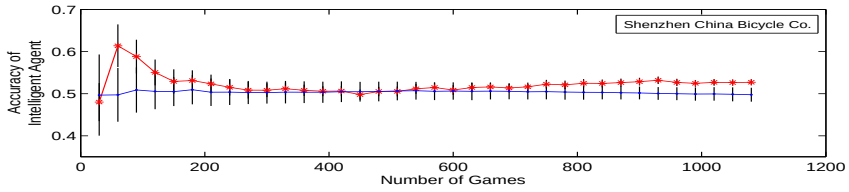
9



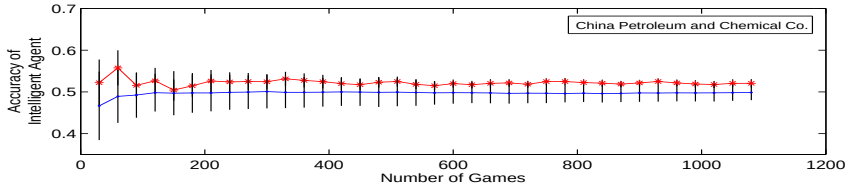
10



11



12



13

Fig. 4. Performance of MGDM in real stock and foreign exchange markets.

Table 2. Descriptions on stock data and the Dollar-RMB exchange rate.

Information / Name	Stock index	Beginning of test date	Mean accuracy after 1100 round
1. The Exchange Rate of RMB against U.S.Dollar	-	Dec 04 2000	63.92%
2. Shanghai Pudong Development Bank Co.	600000	Nov 10 1997	55.90%
3. CITIC Securities Co.	600030	Jan 06 2003	54.40%
4. China Yangtze Power Co.	600900	Nov 18 2003	55.22%
5. Shandong Bohui Paper Industrial Co.	600966	Jun 08 2004	55.81%
6. Kweichou Moutai Co.	600519	Aug 27 2001	53.66%
7. China Minsheng Banking Co.	600016	Dec 19 2000	55.41%
8. China Vanke Co.	000002	Dec 23 1991	53.36%
9. Baoshan Iron & Steel Co.	600019	Dec 12 2000	53.44%
10. Shenzhen Development Bank Co.	000001	Dec 23 1991	53.85%
11. Shenergy Co.	600642	Apr 16 1993	53.61%
12. Shenzhen Chine Bicycle Co.	000017	Apr 01 1992	52.71%
13. China Petroleum and Chemical Co.	600028	Aug 08 2001	52.06%

upper curve in subfigures 2, 3, 4 (Shanghai Pudong Development Bank Co. and CITIC Securities Co. and China Yangtze Power Co.) are almost stable at after 1100 rounds. However, in the last picture, after 1100 rounds, the deviations of the two curves are overlapped, which means that the MGDM is not statistically better than a random walk in this situation. The computation time is about 5 hours by running the Matlab code of the experiment for 30 times with an Intel Pentium dual-core PC.

4 Conclusions

In this paper, we proposed a framework of learning time-series data by considering the collective data is an aggregation of several subgroup behaviors where each group of agents are following the minority game. By using a GA to explore the combination of decomposition structure of the system, an intelligent agent can beat the mark with a small winning rate.

We tested the framework on a few real-world stock prices and Dollar-RMB exchange rate. For most of the cases, the new framework of MGDM performs statistically better than a random walk - that proves the inefficiency of the market. The future work will focus on obtaining the real returns on stock markets.

Acknowledgment

This work is partially funded by the NCET Program of the Chinese Ministry of Education.

References

1. W. B. Arthur, Bounded rationality and inductive behavior (the El Farol problem). *American Economic Review 84*. 406 (1994).
2. D. Challet, M. Marsili, and Y. C. Zhang, Stylized facts of financial markets and market crashes in minority games.*Physica A 294*, 514 (2001).
3. D. Challet, M. Marsili, and R. Zecchina, Statistical mechanics of systems with heterogeneous agents: Minority Games.*Phys. Rev. Lett. 84*, 1824 (2000).
4. D. Challet and Y. C. Zhang, Emergence of cooperation in an evolutionary game.*Physica A 246*, 407 (1997).
5. D. K. Gode and S. Sunder, Allocative efficiency of markets with zero-intelligence traders: Market as a partial substitute for individual rationality, *Journal of Political Economy*, 101(1), 119-137 (1993)
6. J. H. Holland, *Emergence: From Chaos to Order* (1998).
7. G. Li, Y. Ma, Y. Dong, Z. Qin. Behavior learning in minority games, to appear in *Collaborative AgentDResearch and Development International Workshop(CARE)* (2009).
8. N. F. Johnson, P. Jefferies, and P. M. Hui, *Financial Market Complexity*, Oxford University Press, Oxford, (2003)
9. T. S. Lo, P. M. Hui, and N.F. Johnson, Theory of the evolutionary minority game, *Phys. Rev. E 62*, 4393 (2000)
10. R. N. Mantegna and H. E. Stanley, *An Introduction to Econophysics: Correlations and Complexity in Finance*, Cambridge University Press (1999)
11. Z. Qin, Market mechanism designs with heterogeneous trading agents, *Proceedings of Fifth International Conference on Machine Learning and Applications (ICMLA-2006)*, pp. 69-74, Orlando, Florida, USA (2006)
12. Z.Qin, Naive Bayes classification given probability estimation trees, the Proceedings of ICMLA-06, pp. 34-39, (2006)
13. Z. Qin and J. Lawry (2005), Decision tree learning with fuzzy labels, *Information Sciences*, 172/1-2: 91-129 (2005)
14. A. Rapoport, A.M. Chammah and C.J. Orwant, *Prsoner's Dilemma: A Study in Conflict and Cooperation*, Uni. of Michigan Press, Ann Arbor (1965)
15. V. L. Smith, An experimental study of competitive market behavior, *Journal of Political Economy* 70, pp.111-137 (1962)
16. http://en.wikipedia.org/wiki/El_Farol_Bar_problem
17. http://big5.newone.com.cn/download/new_zszq.exe
18. http://bbs.jjxj.org/thread-69632-1-7.html