

Collective game behavior learning with probabilistic graphical models



Zengchang Qin^{a,*}, Farhan Khawar^b, Tao Wan^{c,*}

^a Intelligent Computing and Machine Learning Lab, School of Automation Science and Electrical Engineering, Beihang University, Beijing, China

^b Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong, China

^c Department of Biological Science and Medical Engineering, Beihang University, China

ARTICLE INFO

Article history:

Received 19 August 2015

Received in revised form

25 November 2015

Accepted 25 January 2016

Communicated by Jiahui Qin

Available online 27 February 2016

Keywords:

Minority game

Probabilistic graphical model

Stock index prediction

Collective intelligence

Behavior decomposition

ABSTRACT

The *minority game* is a simple game theory model for describing the collective behavior of agents in an idealized situation where they compete for some finite resources. In this paper, we assume that collective behavior is generated by the aggregation of independent actions of agents and the action follows the minority game. A probabilistic machine learning model is proposed to model the generative process of how collective behavior emerges from individual actions. By training on collective data, we can infer the most likely parameters use the trained system to make predictions. This model can be regarded as a new learning paradigm of analyzing collective data by decomposing the generative process into independent micro-level games. To demonstrate the effectiveness of the model, we conduct experiments on an artificial data set and the real-world data. A set of selected stock indices are tested to capture their rises and falls in the market.

© 2016 Elsevier B.V. All rights reserved.

1. Introduction

Collective intelligence is a shared or group intelligence that emerges from the interactions (both cooperative and competitive) of individual agents and appears in consensus decision making. The collective behavior of systems of many interacting degrees of freedom has been studied intensively in physics [23,22] and different tools are applied in the study of emergence of the collective behavior from interacting agents in economics and sociology [3]. Multi-agent models are one of the most used approaches where each individual agent is self-interested and follows its own rules. Prevalence of cooperation in groups of selfish agents seems to be contradictory to the basic premise of natural selection. Evolutionary game theory provides theoretical framework to address this problem. For example, Szolnoki et al. [29] find that *wisdom of groups*¹ could promote cooperation in evolutionary social dilemmas. Nowak [21] summarizes five rules of the evolution of cooperation and even puts *natural cooperation* as the third fundamental principle of evolution beside mutation and natural selection. Most related research in evolutionary games is focused on the evolution of the system dynamics [23]. For example, the prisoner's dilemma (PD)

game in particular is considered a classical paradigm of cooperation vs defection [20]. Many variants have been proposed by considering the PD game in different spatial structures including ordered networks [30], scale-free networks [16] and small-world networks [13]. A comprehensive review is available in [23]. In this research, we turn our focus on the prediction power of the agent-based system by decomposing the collective behavior into micro-level actions of agents by reconstructing the generative process of collective behavior using a probabilistic graphical model [14].

Multi-agent models are widely used to investigate how cooperative behavior emerges in different research areas including psychology [27], micro-economics [10,28], financial market modeling [11,12] and market mechanism designs [24,25]. In agent-based modeling, the involving agents are with similar capability and compete for a limited resource. Yet, the predictive power of such systems is rather limited: the mapping from the collective behavior to agents' individual actions is a one-to-many. Information is lost in the process of aggregation from individual behavior of many to one macro-level behavior. In order to overcome this problem, Qin et al. [26] assume that the agents are homogeneous with some representative behavior controlled by strategies. By tuning the strategy parameters at agent level, we can find that which strategy may yield the most likely macro-level behavior.

If we assume that every agent in the market knows the history data, the key problem is to how to decide to act based on this global information. These agents may share global information and learn from past experience. In this research, we assume that

* Corresponding authors.

E-mail addresses: zcqin@buaa.edu.cn (Z. Qin), fkhawar@ust.hk (F. Khawar), tao.wan.wan@gmail.com (T. Wan).

¹ Wisdom of groups (or wisdom of crowds) is the collective opinion of a group of individuals rather than that of a single expert.

agents act based on a simple game that is referred to as the *minority game*. Proposed by Challet and Zhang [4], the minority game (MG) is a simplified version of the El-Farol Bar [1] problem: In the minority game, each odd number of players must choose one of the two choices independently at each turn, the players who end up in the minority side win the game. In the minority game, every agent is assigned with strategy which the agent uses to make choice based on perceived global information (collective behavior) [7]. Heterogeneity is important as it is unrealistic that all agents follow the same deterministic strategy [15].

In previous works [18,26], collective behavior is assumed to be generated by groups of agents with the same strategies. The Genetic Algorithm (GA) is used to optimize the agent behavior parameters in order to get the most likely guess of the collective behavior. Also, different games can be used to model the individual behavior. More accurate a model is for individual behavior, more likely collective behavior we can obtain from their aggregation [8]. In this paper, we propose a novel generative graphical model for modeling the process of generation of the collective behavior from the individual behavior. We use the new model to infer the behavior of individual agents from the available global information. The trained system can be used to predict the future collective behavior. The main contributions of this paper are the following: (1) By studying the relationship between collective behavior and its decomposed agent behavior that can be modeled by the minority game, we propose a new framework for data mining. (2) A graphical model is proposed to model the generative process of the collective data from a group of individual actions. Based on this model, we can infer the individual strategies and predict future collective behavior. (3) We apply this framework to practical time-series data mining tasks including FX rate and stock index prediction. The experimental results demonstrate the effectiveness of the new proposed model.

This paper is organized as follows: Section 2 introduces the minority game and how to use it to model agent behavior. In Section 3, we propose a graphical model for collective behavior learning. Inference and prediction of the model is given in Section 4. Section 5 gives experimental results on artificial data as well as the real market data. Finally, conclusions and future work are given in Section 6.

2. Behavior modeling using the minority game

The minority game (MG) was originated from the El Farol Bar problem and was formulated to analyze decision-making [1]. In each round of the game, there are an odd number of players and each one must independently choose one of the two choices, the players who end up on the minority side are winners. The choice of the minority players is referred to as the winning choice. There is no prior communication among players and we assume that the only global information available is the number of players of two choices from the previous rounds.

Let us first introduce the notations and terminology used in this paper:

Agent: A player of the game is referred to as an *agent* and it is the entity that makes decisions based on its *strategy*. The number of agents that participate in the game is denoted by N , which should be an odd number. An agent is indexed by an integer A : $A \in \{1, 2, \dots, N\}$.

Choice: An action made by an agent: a choice C has two possible values: $C \in \{0, 1\}$. The total number of choices equals to N in each round of the game.

Game: In a minority game, choices are represented by a vector of N elements of binary values $[C_1, C_2, \dots, C_N]$ where $C_i \in \{0, 1\}$ the

i th agent's choice in this game. The total number of games is denoted by G .

Minority choice: In every game, the choice of the agents is on the minority side. Formally, let t denote the current game, $t \in \{1, 2, \dots, G\}$, then the minority choice in game t is redefined by:

$$r(t) = \begin{cases} 0 & \text{If } \sum_{n=1}^N C_n(t) > \frac{N}{2} \\ 1 & \text{Otherwise} \end{cases} \quad (1)$$

Memory and history: In the minority game, we assume that the agent's actions are governed by its strategy and minority choices of previous rounds of the game. An agent with m -bit memory means that the agent will take the information (minority choices) of the previous m rounds into account for making its current decision. The minority choices of the last m rounds at time t can be simplified to the *history*:

$$H(t) = \gamma([r(t-m), \dots, r(t-2), r(t-1)]) \quad (2)$$

where $\gamma(\cdot)$ converts a binary vector into a decimal number² for representational convenience. $H \in \{1, 2, 3, \dots, K\}$ and $K = 2^m$, where K is the maximum number of all possible histories.

Strategy: An agent's *strategy* can be represented by a set of rules, a lookup table or a function, which takes the histories as input and output an action of the agent (i.e., make a choice) [4–6]. A strategy, denoted by S , can be regarded as a particular set of choices on all the permutations on previous history of minority choices. Therefore, there are 2^{2^m} possible strategies in the strategy space and we assume that at one time each agent has exactly one strategy. For example, Table 1 shows the 3-bit memory, history and a sample strategy. Considering an agent employing the strategy S , when given memory is $(101)_2$ (with corresponding history 6), the agent will definitely choose the side of 0.

Probabilistic strategy: As we can see from the above example, the strategy can be represented by a lookup table and deterministic. It is not a true assumption when considering the real-world choice making. In this research, we first propose the *probabilistic strategy* (PS), which maps the history to a probability distribution over the two choices instead of one choice only. Therefore, a PS is a set of distributions (one distribution for each given history) from which the agent's choice can be sampled. We can use a Bernoulli distribution $P_B(\cdot)$ for choosing from two choices. The bottom row of Table 1 shows a sample of probabilistic strategy, where $P_B(\alpha)$ gives the probability of choosing 0, and of course, the probability of choosing 1 is $1 - P_B(\alpha)$.

The advantage of using probabilistic strategy is that we are able to incorporate uncertainty in the learning process by placing a prior over choices. Based on the observations at each round of the game, we can update the posterior in order to estimate the probabilistic strategy. In the case of the minority game where we have only two possible choices, therefore, we can use the Beta distribution³ as the prior.

3. Graphical model for collective behavior learning

Previous works [8,15,18] show that collective behavior can be decomposed into the aggregation of individual agents' actions and each agent has its own deterministic strategy. However, in the real-world, there is always uncertainty in decision making. It is

² For the mathematical convenience, $\gamma(\cdot)$ actually gives the output value by adding 1 to the corresponding decimal number of the binary memory.

³ Beta distribution can be regarded as a special case of Dirichlet distribution. If the game is with more than $N(N \geq 3)$ choices, we then need to use N -dim Dirichlet distribution as the prior.

Table 1
A sample of strategy (S) and probabilistic strategy (PS) associated with 3-bit memory and the corresponding history (H).

Memory	(000) ₂	(001) ₂	(010) ₂	(011) ₂	(100) ₂	(101) ₂	(110) ₂	(111) ₂
History (H)	1	2	3	4	5	6	7	8
Strategy (S)	1	0	1	1	0	0	0	1
PS	$P_B(0.1)$	$P_B(0.8)$	$P_B(0.3)$	$P_B(0.1)$	$P_B(0.6)$	$P_B(0.8)$	$P_B(0.6)$	$P_B(0.3)$

hardly feasible for agents to follow deterministic strategies. In this paper, we assume that the collective behavior is generated by agents with probabilistic strategies.

Probabilistic graphical models (or simply graphical models) [2] have the capability to deal with uncertainty by incorporating prior beliefs about the domain and updating these beliefs as new evidence is obtained. Using graphical models we can construct richly structured models to understand hidden relations. In this research, we use a graphical model to describe the generative process of the collective behavior. The task of the graphical model is two-fold: (1) to decompose the collective behavior by inferring individual agent behaviors; (2) to use these learned individual behaviors to predict the future collective behavior.

3.1. Generative process

In this model, the minority choice and history are observed, while agent index, choice distribution and agent distribution are unobserved. As shown in Fig. 1, where the rectangular “plates” denote replication and the number at bottom right of the box is the times of replications. The generative process in the bigger plate at bottom is replicated N times, therefore for each of the N choices we will sample an agent distribution, agent index and a choice from the choice distribution, corresponding to the observed history. The smaller plate above represents repeated sampling of choice distribution for every agent given on all possible histories.

We start by choosing a probabilistic strategy from a Dirichlet prior. Given the probabilistic strategy and the current history, we can choose a particular distribution, from which we can sample a choice. The agent who is chosen to make this choice is sampled based on an agent distribution. This is repeated N times to generate all the choices of a game.

- For each agent with a given history:
 - Choose the *choice distribution* from a Dirichlet prior: $\phi_{n,k} \sim \text{Dir}_2(\beta)$.
- Observe the history H for the current game.
- For each choice:
 - Choose the *agent distribution* from a Dirichlet prior: $\psi_n \sim \text{Dir}_N(\alpha)$.
 - Choose an agent index $A_n \sim \text{Mult}(\psi_n)$, $A_n \in \{1, \dots, N\}$.
 - Sample a choice from the choice distribution given the agent index and history, i.e. $C_n \sim \text{Bernoulli}(\phi_{A_n,H})$, $C_n \in \{0, 1\}$.

where $\text{Mult}(\cdot)$ denotes a discrete multinomial distribution. We use the concentration parameters α and β to parameterize the symmetric prior Dirichlet distributions.

The choice distribution $\phi_{n,k}$ is a probability distribution on two possible choices for agent $n \in \{1, 2, \dots, N\}$ given history $k \in \{1, 2, \dots, K\}$. For a single agent n , the set of choice distributions $\{\phi_{n,1}, \phi_{n,2}, \dots, \phi_{n,K}\}$ is just the probabilistic strategy of agent n . Each $\phi_{n,k}$ is sampled from a sparse symmetric Dirichlet (Beta) distribution parameterized by β . Given that the history is 1 and the current choice in the game is made by agent 2 (agent index $A=2$), $\phi_{1,2}$ is used to represent the probability distribution of choosing 1 or 0. The distribution ψ_n , $n \in \{1, 2, \dots, N\}$, is the agent distribution for each choice C_n and it is an N -dimensional random variable where each component of the vector gives the probability of selecting the agent

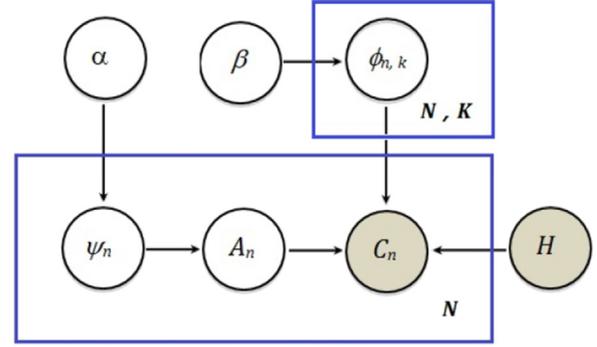


Fig. 1. Graphical model for collective behavior decomposition using the minority games. Shaded random variable C and H can be observed while all others are latent variables.

index associated with that component. This agent distribution is sampled from another Dirichlet distribution parameterized by α . Generally, for each choice C_n made in a game we have the same history H and a different agent distribution ψ_n so that the same agent is not selected for making every choice in the game.

$$\psi_u | H(t) \neq \psi_v | H(t) \quad \text{where } u, v \in \{1, 2, \dots, N\}$$

Also the n th choice of another game with a different history will have a different agent distribution since the agent distribution depends on the history:

$$\psi_n | H(t-1) \neq \psi_n | H(t) \quad \text{where } H(t-1) \neq H(t)$$

Similar to choice distribution, we save all the agent distributions in a 2-dimensional array of vector elements where the rows correspond to the order of the choice in the game that is to be made i.e. the 1st, 2nd, ..., N th choice and the columns correspond to the history. Since every game will have the same history, one column of this array will contain all the agent distributions for that game.

It is worth mentioning that the choices C_n are unordered choices. If the actual choices made by the $N=5$ agents were $[1, 0, 1, 0, 0]$ in one game and $[0, 1, 1, 0, 0]$ in the other, then in both cases the C_n would be $[0, 0, 0, 1, 1]$. This is because we want to use our model to decompose the collective behavior that only provides global information of how many agents made a particular choice, but we do not care not exactly which individual made that choice. Once we have the information of how many agents made the choice, it is enough to construct the model.

3.2. Joint distribution of the model

The new model we discussed above is a generative graphical model which explores the relationship between micro-level actions and macro-level behavior. The dependency relation can be clearly seen from Fig. 1, therefore, the joint distribution of the new model is as follows:

$$\begin{aligned}
 p(\psi, A, \phi, C, H) &= p(\psi_{1:N}, A_{1:N}, \phi_{1:N,1:K}, C_{1:N}, H | \alpha, \beta) \\
 &= \prod_{i=1}^N \prod_{k=1}^K p(\phi_{i,k} | \beta) p(H) \\
 &\quad \times \left[\prod_{n=1}^N p(\psi_n | \alpha) p(A_n | \psi_n) p(C_n | A_n, \phi_{1:N,1:K}, H) \right] \quad (3)
 \end{aligned}$$

Table 2

Experiment results on an artificial dataset: the training data is generated by giving deterministic strategies, the learnt probabilistic strategies using the new graphical model are right under its corresponding deterministic strategies.

Agent	Strategy	Memory/History							
		(000) ₂ 1	(001) ₂ 2	(010) ₂ 3	(011) ₂ 4	(100) ₂ 5	(101) ₂ 6	(110) ₂ 7	(111) ₂ 8
1	Original	0	0	1	0	1	1	0	0
	Learned	$P_B(0.92)$	$P_B(0.93)$	$P_B(0.07)$	$P_B(0.92)$	$P_B(0.08)$	$P_B(0.07)$	$P_B(0.95)$	$P_B(0.95)$
2	Original	0	0	0	1	0	0	1	1
	Learned	$P_B(0.91)$	$P_B(0.93)$	$P_B(0.93)$	$P_B(0.08)$	$P_B(0.92)$	$P_B(0.92)$	$P_B(0.05)$	$P_B(0.05)$
3	Original	1	1	0	1	0	1	0	0
	Learned	$P_B(0.09)$	$P_B(0.07)$	$P_B(0.93)$	$P_B(0.08)$	$P_B(0.92)$	$P_B(0.07)$	$P_B(0.95)$	$P_B(0.94)$
4	Original	1	1	1	0	1	0	0	1
	Learned	$P_B(0.08)$	$P_B(0.07)$	$P_B(0.07)$	$P_B(0.46)$	$P_B(0.08)$	$P_B(0.71)$	$P_B(0.67)$	$P_B(0.06)$
5	Original	Rand.							
	Learned	$P_B(0.41)$	$P_B(0.50)$	$P_B(0.60)$	$P_B(0.92)$	$P_B(0.50)$	$P_B(0.93)$	$P_B(0.95)$	$P_B(0.49)$

Table 3

Equivalent strategies: table shows strategies of 6 agents and each group of three agents results in the same minority choice.

Agent Index A	History H			
	(00) ₂ 1	(01) ₂ 2	(10) ₂ 3	(11) ₂ 4
1	0	0	1	0
2	0	1	0	0
3	1	0	0	0
Minority Choice r	1	1	1	1
4	1	1	0	0
5	0	0	0	0
6	0	0	1	0
Minority Choice r	1	1	1	1

where $\phi_{1:N,1:K}$ denotes the set of choice distributions

$$\{\phi_{n,k} | 0 < n \leq N, 0 < k \leq K\}$$

This is to say that the probability of choose C_n is evaluated by the probabilistic strategy of A_n given the observed history H :

$$p(C_n | A_n, \phi_{1:N,1:K}, H) = \phi_{A_n, H} \quad (4)$$

Notice that since H is observable and does not depend on other variables, it is deterministic and $p(H)$ can be omitted from Eq. (4). We can also see that once choices are observed the agent distributions $\psi_{n,H}$ and choice distributions $\phi_{n,k}$ are conditionally dependent according to d -separation.

4. Model inference and prediction

In generative probabilistic modeling, we treat the data as arising from a generative process that includes hidden and observed variables. This generative process defines a joint probability distribution over both the observed and latent random variables. We perform data analysis by using the joint distribution to compute the conditional distribution of the latent variables. Our goal is to infer the individual behaviors, then use these individual behaviors to predict the choices for the next game and calculate our accuracy of prediction.

4.1. Decomposition of collective behavior

The decomposition of collective behavior can be done by inferring the posterior distribution of the hidden variables. The choices and the history are the macro-level (collective) behavior and the posterior of the agent distributions $p(\psi_{1:N} | C_{1:N}, H)$ and choice

distributions $p(\phi_{1:N,1:K} | C_{1:N}, H)$ are the micro-level (agent) behavior. Here the notation $\psi_{1:N}$ denotes the set of distributions $\{\psi_n | 0 < n \leq N\} = \{\psi_1, \psi_2, \dots, \psi_N\}$. The posterior distribution can be written as:

$$p(\psi_{1:N}, A_{1:N}, \phi_{1:N,1:K} | C_{1:N}, H, \alpha, \beta) = \frac{p(\psi_{1:N}, A_{1:N}, \phi_{1:N,1:K}, C_{1:N}, H | \alpha, \beta)}{p(C_{1:N} | \alpha, \beta)} \quad (5)$$

where the numerator of Eq. (5) is defined in Eq. (3). In order to normalize this posterior distribution we need to marginalize over the latent variables to give the denominator as shown below:

$$p(C_{1:N} | \alpha, \beta) = \prod_{i=1}^N \prod_{k=1}^K \int \prod_{n=1}^N \int p(\psi_n | \alpha) \times \sum_{A_n} p(A_n | \psi_n) p(C_n | A_n, \phi_{1:N,1:K}, H) d\psi_n p(\phi_{i,k} | \beta) d\phi_{i,k} \quad (6)$$

Algorithm 1. Model inference and prediction.

Parameters: α, β , Number of agents N , Number of games G

- 1: Construct a Bayesian inference engine E using Variational Message Passing in Infer.Net
- 2: **for** $t = 1 \rightarrow G$ **do**
- 3: **if** $t = 1$ **then**
- 4: Assign symmetric Dirichlet priors as in Eq. (7)
- 5: **else**
- 6: Assign posterior distributions from the last game as the current prior distribution as in Eq. (8)
- 7: **end if**
- 8: Observe choices $C_{1:N}(t)$ and history $H(t)$ and input them to E .
- 9: Execute E and calculate the posterior agent and choice distributions.
- 10: Given the posteriors of step-9 and $H(t+1)$, execute E to infer predicted choice distributions of Eq. (9)
- 11: To get $PC_{1:N}(t+1)$, sample from the distributions in Eq. (9)
- 12: Predict the minority choice of the next game $\hat{r}(t+1)$ from Eq. (11)
- 13: Calculate the prediction accuracy from Eq. (12)
- 14: **end for**

Posterior inference can be done using approximate message passing algorithm [14], specifically we use the Variational Message Passing (VMP) algorithm [31] in the Infer.Net [19] package. VMP is deterministic approximate inference algorithm that is guaranteed to converge to some solutions and it works by using only local

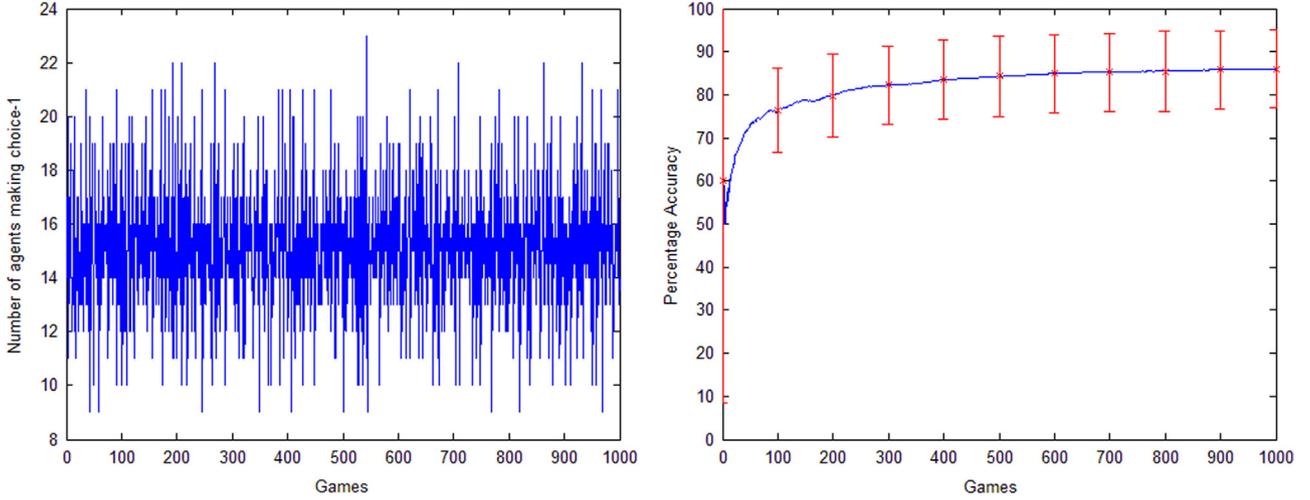


Fig. 2. Prediction accuracy on artificially generated data with random agents. Left-hand side (random collective behavior): the number of agents making choice-1 in 1000 runs of fixed strategy minority games. Right-hand side: the accuracy of the prediction in percentage (%).

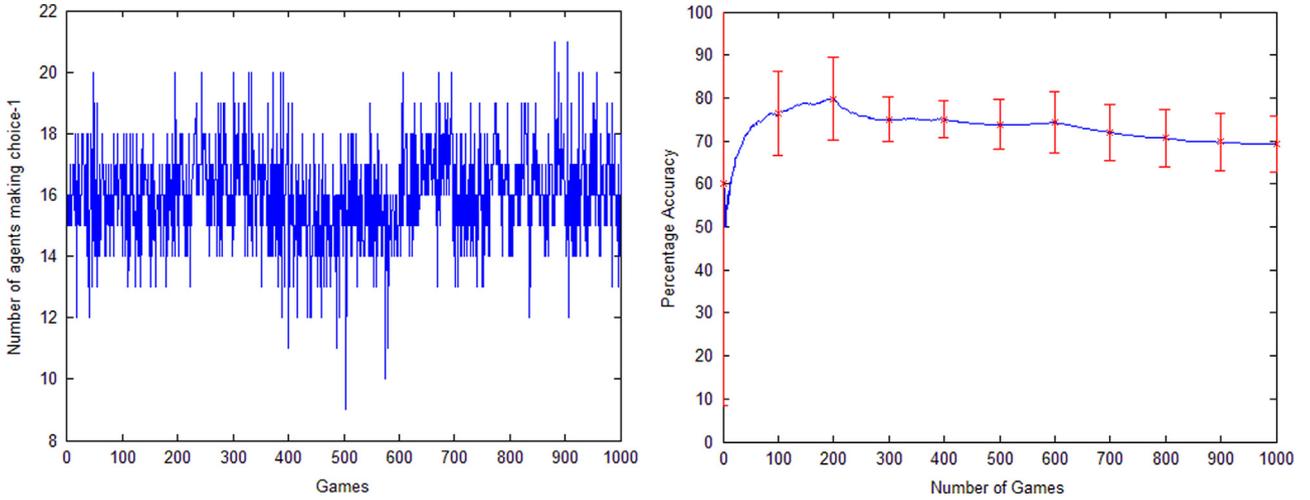


Fig. 3. Prediction accuracy on artificially generated data with adaptive and random agents. Left-hand side (random collective behavior): the number of agents making choice-1 in 1000 runs of fixed strategy minority games. Right-hand side: the accuracy of the prediction in percentage (%).

message passing operations. The inference algorithm is outlined in Algorithm 1.

We start by constructing a Bayesian inference engine E using Variational Message Passing in Infer.Net. For the first game, we start by assigning symmetric Dirichlet priors to agent distribution and choice distribution:

$$\begin{aligned} \psi_{1:N}(1) &\sim \text{Dir}_N(\alpha) \\ \phi_{1:N,1:K}(1) &\sim \text{Dir}_2(\beta) \end{aligned} \quad (7)$$

Otherwise, for subsequent games we start by assigning the posterior distributions from the last game as the current prior:

$$\begin{aligned} p(\psi_{1:N}(t)) &= p(\psi_{1:N}(t-1) | C_{1:N}(t-1), H(t-1)) \\ p(\phi_{1:N,1:K}(t)) &= p(\phi_{1:N,1:K}(t-1) | C_{1:N}(t-1), H(t-1)) \end{aligned} \quad (8)$$

where $C_{1:N}(t-1)$ are the observed choices and $H(t-1)$ is the observed history of the previous game $t-1$, $p(\psi_{1:N}(t))$ and $p(\phi_{1:N,1:K}(t))$ are the priors for the current game t , $p(\psi_{1:N}(t-1) | C_{1:N}(t-1), H(t-1))$ and $p(\phi_{1:N,1:K}(t-1) | C_{1:N}(t-1), H(t-1))$ are the posteriors from the last game. After assigning the priors we observe the collective data for the current game i.e. choices $C_{1:N}(t)$ and history $H(t)$. This collective data can be thought of as an input to the model. Then approximate inference is performed using VMP

and Infer.Net by executing the inference engine E to infer the local behavior which is the posterior of the choice distribution $p(\phi_{1:N,1:K}(t) | C_{1:N}(t), H(t))$ and the agent distribution $p(\psi_{1:N}(t) | C_{1:N}(t), H(t))$.

4.2. Prediction

Once we have obtained the local behavior we can use this local behavior to predict the collective behavior of the subsequent games. Given the inferred posteriors of agent and choice distribution and history of next game $H(t+1)$, we execute the engine E to infer Bernoulli distributions of the *predicted choices* (PC) of the next game which are defined by:

$$\begin{aligned} p(PC_{1:N}(t+1) | \psi_{1:N}(t), A_{1:N}(t+1), \phi_{1:N,1:K}(t), \\ H(t+1)) &\sim \text{Bernoulli}(\phi_{A_{1:N}(t+1), H(t+1)}) \end{aligned} \quad (9)$$

where $PC_{1:N}(t+1)$ are the predicted choices of the next game. To get the predicted choices of the next game we sample from the inferred predicted choice distributions of Eq. (9), therefore:

$$PC_{1:N}(t+1) \sim \text{Bernoulli}(\phi_{A_{1:N}(t+1), H(t+1)}) \quad (10)$$

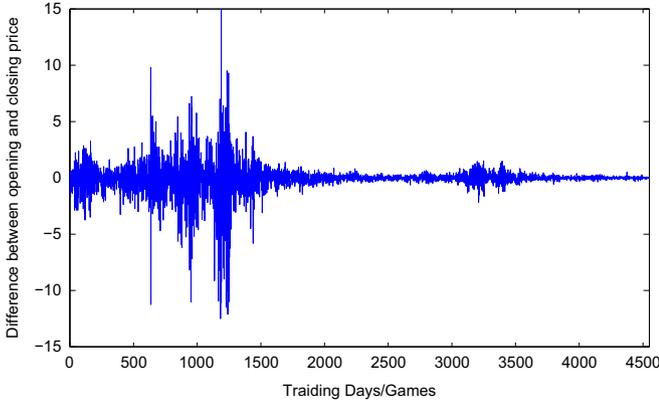


Fig. 4. Price differences between opening and closing prices within a trading day on NOK.

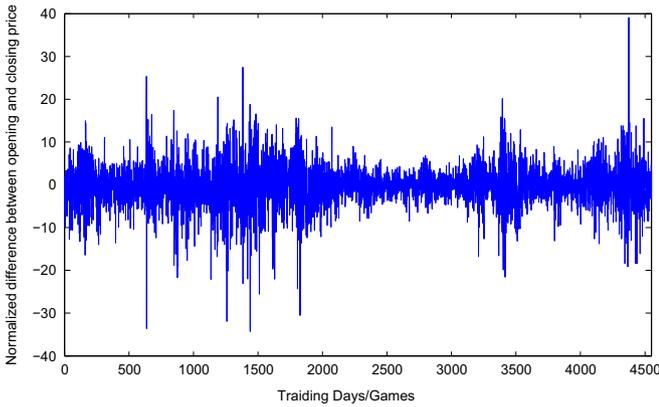


Fig. 5. Normalized price differences between opening and closing prices.

where $PC_{1:N}(t+1) \in \{0, 1\}$. Using these predicted choices we obtain the predicted minority choice of the next game $\hat{r}(t+1)$ defined as:

$$\hat{r}(t+1) = \begin{cases} 0 & \text{If } \sum_{n=1}^N PC_n(t+1) > \frac{N}{2} \\ 1 & \text{Otherwise} \end{cases} \quad (11)$$

We can then use this predicted minority choice to calculate the prediction accuracy $Acc(t)$ after observing game t , which is defined as:

$$Acc(t) = \frac{\#(\hat{r}(t+1) = r(t+1))}{\#(\hat{r}(t+1) = r(t+1)) + \#(\hat{r}(t+1) \neq r(t+1))} \quad (12)$$

where the $\#(\cdot)$ is an incremental counter, initialized with 0 at $t = 1$, that increments by 1 each time its argument (\cdot) is true. In other words, $Acc(t)$ is the number of times the correct prediction of the minority choice is made divided by the total number of games given the data of all the previous games.

5. Experimental Studies

In this section we conduct two experiments to test performance of the new proposed model, one experiment is on an artificial data and the other one is on the real-world financial data. Specifically, we use the model to predict the trend of rises and falls of a stock in market. In our experiments, $\alpha = 0.009$, $\beta = 1$ to ensure uniform symmetric Dirichlet distribution. The reason for choosing $\alpha \ll 1$ is that it makes the agent distribution sparse and we can select only one agent to make that choice. Also for all subsequent

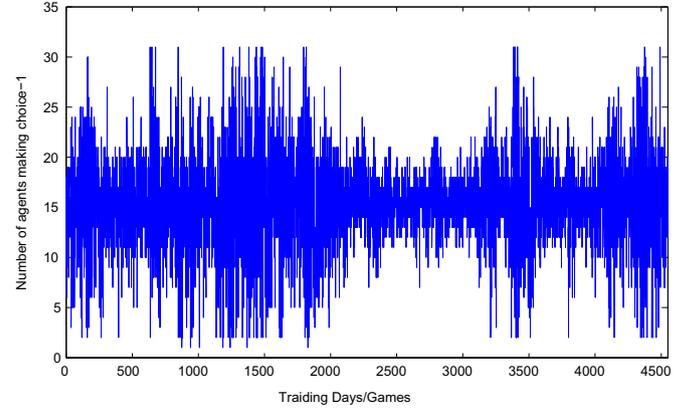


Fig. 6. Estimated numbers of agents making choice 1 in the minority game modeling on the NOK dataset.

experiments we have set $m=4$ and therefore we have used a 4-bit history and $K=16$.

5.1. Experiments with artificial data

To test the validity of our model we first conduct experiments on artificial datasets which are generated according to predefined parameters for the minority game and we can use our model to learn the individual behavior and then make predictions. The artificial data is generated by Algorithm 2. It basically simulates how the collective data is generated by aggregation of agent behavior.

Algorithm 2. Artificial data generation with random agents.

Parameters: Number of agents N , Number of games G , Number of random agents R , History size K , Current history H , Counter C .

- 1: **for** $n = 1 \rightarrow N$ **do**
- 2: Randomly generate a deterministic strategy for each agent.
- 3: **end for**
- 4: **for** $t = 1 \rightarrow G$ **do**
- 5: **for** $n = 1 \rightarrow N$ **do**
- 6: **if** $n > N - R$ **then**
- 7: Make a random choice.
- 8: **else**
- 9: Make a choice based on Agent n 's strategy given H .
- 10: **end if**
- 11: If agent n made choice-1 then increment counter C .
- 12: **end for**
- 13: **if** $C > N/2$ **then**
- 14: Minority choice is 0 and update H .
- 15: **else**
- 16: Minority choice is 1 and update H .
- 17: **end if**
- 18: **end for**

We assume that there are two kinds of agents in the system: The first kind are the agents with strategy (we call them *strategic agents* in order to differentiate them from other types of agents), they make choices governed by its strategy given the observed history. The second kind are *random agents* who follow no strategy at all, they make random decision and that is also why we give them that name. In this experiment, the random agents make choice from a uniform distribution. The presence of random agents is a realistic assumption, because in the real-world, there are always certain unknown factors that make some people take

Table 4
Descriptions of datasets and corresponding prediction accuracy using our model.

Dataset (Index)	Start Date	Acc. $t=800$	Acc. $t=1000$	τ
1. Adidas AG (ADS.DE)	01/03/2000	53.51	53.51	4.00
2. Agilent Technologies Inc (A)	11/10/2003	54.05	54.24	1.50
3. Apple Inc (AAPL)	08/27/1986	56.22	55.52	1.50
4. AT&T Inc(T)	06/30/1988	54.35	53.59	2.50
5. BT Group plc (BT)	04/05/1991	56.86	55.06	1.50
6. General Electric Company (GE)	01/02/1997	53.75	53.05	1.50
7. Google Inc. (GOOG)	05/22/2009	49.46	50.54	0.25
8. Hollywood Media Corp. (HOLL)	04/18/2000	56.22	55.58	2.50
9. Intel Corporation (INTC)	07/09/1986	54.95	56.33	1.00
10. McDonald's Corp. (MCD)	09/27/1991	53.08	55.26	2.00
11. Mercer International Inc. (MERC)	01/07/1991	62.11	59.04	5.00
12. Microsoft Corporation (MSFT)	03/13/1986	51.14	52.73	1.00
13. Nike Inc. (NKE)	06/22/2007	51.91	52.77	1.50
14. Oracle Corporation (ORCL)	07/06/1990	53.48	53.09	1.00
15. Pepsico Inc. (PEP)	10/09/2001	54.52	53.49	3.00
16. The Boeing Company (BA)	01/02/1962	54.25	52.93	1.50
17. The Walt Disney Company (DIS)	04/29/1986	55.15	53.55	15.5
18. Toyota Motor Corporation (TM)	04/13/1993	69.30	67.39	4.00
19. Wal-Mart Stores Inc. (WMT)	08/25/1987	55.85	56.49	2.00
20. Forest Laboratories Inc. (FRX)	04/19/1988	56.45	55.56	1.50
21. Nokia Corporation (NOK)	04/25/1995	51.67	52.91	1.00
22. Sonic Corp. (SONC)	02/28/1991	54.78	55.44	5.00
23. CN Eastern Airlines (600115SS)	11/05/1997	52.31	51.29	8.00
24. CN Medicine & Health (600056SS)	09/16/2005	51.61	52.15	4.00
25. CN Minsheng Banking (600016SS)	12/19/2000	56.59	56.14	15.0
26. CN Television Media (600088SS)	06/16/1997	49.43	50.82	5.00
27. CN Unicom (Hong Kong) (600050SS)	10/09/2002	54.45	54.54	15.0
28. CN Yangtze Power (600900SS)	11/18/2003	52.17	52.43	20.0
29. Huaneng Power (600011SS)	12/06/2001	50.27	50.70	25.0
30. Qingdao Haier (600690SS)	01/02/2001	50.74	50.44	20.0
31. Shandong Bohui Paper (600966SS)	06/08/2004	54.52	52.29	20.0
32. SPD Bank Co., Ltd. (600000SS)	01/02/2001	51.54	51.85	35.0
33. Shenergy Company (600642SS)	01/02/2001	53.95	51.05	8.00
34. Hisense Electric (600060SS)	04/22/1997	49.10	50.52	25.0

random decisions. Even some people's decisions are totally random and unpredictable.⁴

In our experiment, there are a total of R random agents and $N - R$ strategic agents. We start by generating a strategy for each of the strategic agents. These strategies were generated by sampling from a uniform distribution over the closed integer interval of $[0, 1]$ for every value of history and hence the length of each strategy is equal to K . The agent makes choice from its own strategy and the remaining R random agents make random choices. After every agent makes its independent choice at the end of the game, the statistics of choices-1 and 0 can be obtained, by which we can easily find which are the minority choices and which agents are winners in this round.

5.1.1. Learning with complete information and fixed strategies

In this experiment, we assume that the complete information of agents' actions is available, i.e., we know which agent has made which choice. Using this information we can use the graphical model to infer the strategies of the agents. Unordered choices of 5 agents for 100 games and the distributions ψ_n are input to the graphical model, where one of the five agents is a random agent. Table 2 shows the strategies used to generate the input data and the corresponding strategies learned by the graphical model (Agent 5 is the random agent).

For every agent the original strategies are represented as a choice of either 0 or 1 for every 3-bit history and the learned strategies are represented by a Bernoulli distribution $P_B(\alpha)$. From Table 2 we can see

⁴ Someone may argue that such uncertainty is due to the lack of information, it does not really mean that this human is random and unpredictable. In our experiments, we simply consider the agents with no predefined deterministic strategy are random agents. We do not try to go deep into modeling their random patterns.

that, with complete information and 20% random agents, the model can learn the strategies with a fairly high accuracy. The only exception is for the 4th agent, it fails to obtain a good approximation of the original strategy given history 4 (memory $(011)_2$). It is also obvious that there are bad results for the random agent, as no system can perform better than 0.5 in our case.

5.1.2. Learning with partial information and fixed strategies

Learning with complete information is however unrealistic as in practice we might only be given the choices of the agents and the corresponding history and the information of which agent made what choice (the distributions ψ_n) might not be available. Although, in this scenario learning the strategies of agents exactly might not be possible, we would still be able to learn strategies that collectively give the same result. This is because the strategy space grows exponentially with m and the mapping from the strategy space to the minority choice r is not unique. For example two possible equivalent groups of strategies for three agents are shown in Table 3 and it can be seen that the choices of agents 1, 2, 3 and agents 4, 5, 6 result in the same global behavior r . Therefore, once the minority choices $\{1, 1, 1\}$ are provided to the graphical model, it can learn any of the equivalent strategies of the agents that would result in these minority choices.

Therefore, the choices and the history of each game are observed and the corresponding strategies are learned. In the following experiment, we set the total number of agents as 31 and 10% of these are random agents in 1000 games. We repeat this experiment 10 times and the average accuracy (Eq. (12)) is plotted with error bars in Fig. 2. We can see that the prediction accuracy can reach up to almost 86% on average after 1000 runs of the minority game even though the global behavior seems totally random and unpredictable.

5.1.3. Learning with adaptive strategies

In the previous section, we assume that all agents keep fixed strategies for all the time. This could be unrealistic when an agent receives little reward with its initial strategy. Therefore, in the following experiments, we allow some agents to change their strategies when they fail to obtain fair rewards. This can be measured by the *losing probability* that indicates the percentage that agent loses in the past games. There must exist a threshold λ for an adaptive agent: if its losing probability is bigger than λ , it will change its strategy by regenerating a random strategy; otherwise, it will keep the same. We assume that only a subset of agents are capable of changing their strategy by using feedbacks. Then, we have a mixture of agents of being *adaptive* (able to re-evaluate its strategy and change it), *strategic* (with fixed strategy and never change) and *random*.

In the following experiments, we assume that each adaptive agent has its own threshold sampled from the normal distribution: $\lambda \sim \mathcal{N}(0.6, 0.12)$. This configuration can generate heterogeneous behavior to be more realistic to the real-world scenarios. Moreover, the strategy change can only happen after every 200 games as this allows the model to observe sufficient data to make valid predictions.

Fig. 3 shows the performance of the model in 1000 runs, in which there are 31 agents with 10% random agents and 20% adaptive agents. The averaged accuracy of 10 games is plotted along with error bars in the right-hand side of Fig. 3. As seen from the results, the accuracy of the prediction increases to around 80% within the first 200 games after that the adaptive agents start to change their strategy, and that results in a sudden drop in prediction accuracy. Subsequently, after every 200 games the strategies will change. Surprisingly, the model can quickly capture the change and reach a pretty high accuracy of 70%, though it is slightly worse comparing to the experiments with strategic agents only.

5.2. Experiments on stock market

The minority game is related to many real-world complex collective behavior [4,12,17] including financial markets. On the macro-level, the behavior of real-world market data appears random and unpredictable, but in our experiments we assume the collective behavior of these markets as being generated according to the minority game and then by using the new proposed model, we can infer local behaviors to predict possible future trends of these market data.

5.2.1. Preprocessing of stock market data

In our experiment, we ignore the exact price change of a stock but only care whether the stock rises or falls within a trading day. That can be regarded as two choices in the minority game. Formally, for a stock market index, let the opening price of that trading day be denoted by $P_o(t)$ and the closing price for that trading day be denoted by $P_c(t)$, then the stock index fluctuation for a trading day can be encoded to the minority choices by:

$$r(t) = \begin{cases} 1 & \text{If } P_c(t) \geq P_o(t) \\ 0 & \text{Otherwise} \end{cases} \quad (13)$$

From these minority choices we can use Eq. (2) to obtain the history value for each game which will be observed by our graphical model. For the following experiments we arbitrarily decide $N = 31$ because we can never know the actual number of agents in the real market so we decide on a value that leads to fast execution while not compromising on performance. Actually, arbitrary choice of N is a valid assumption because for any large number of agents it is possible to reduce their actions to lesser number group

behavior. One agent is not a trader, but it is a group of traders with similar behaviors.

Based on the fluctuation of stock prices, we can simulate the relative numbers of choice 1 and 0. Fig. 4 shows the price change within a trading day on Nokia Corporation (NOK). As we can see from the figure, the scale of price difference is drastically changed after trading day 1500. Therefore, we need to normalize the data in order to obtain a fair estimation of parameters. We take the maximum of the opening price P_{max} (rounded to the nearest

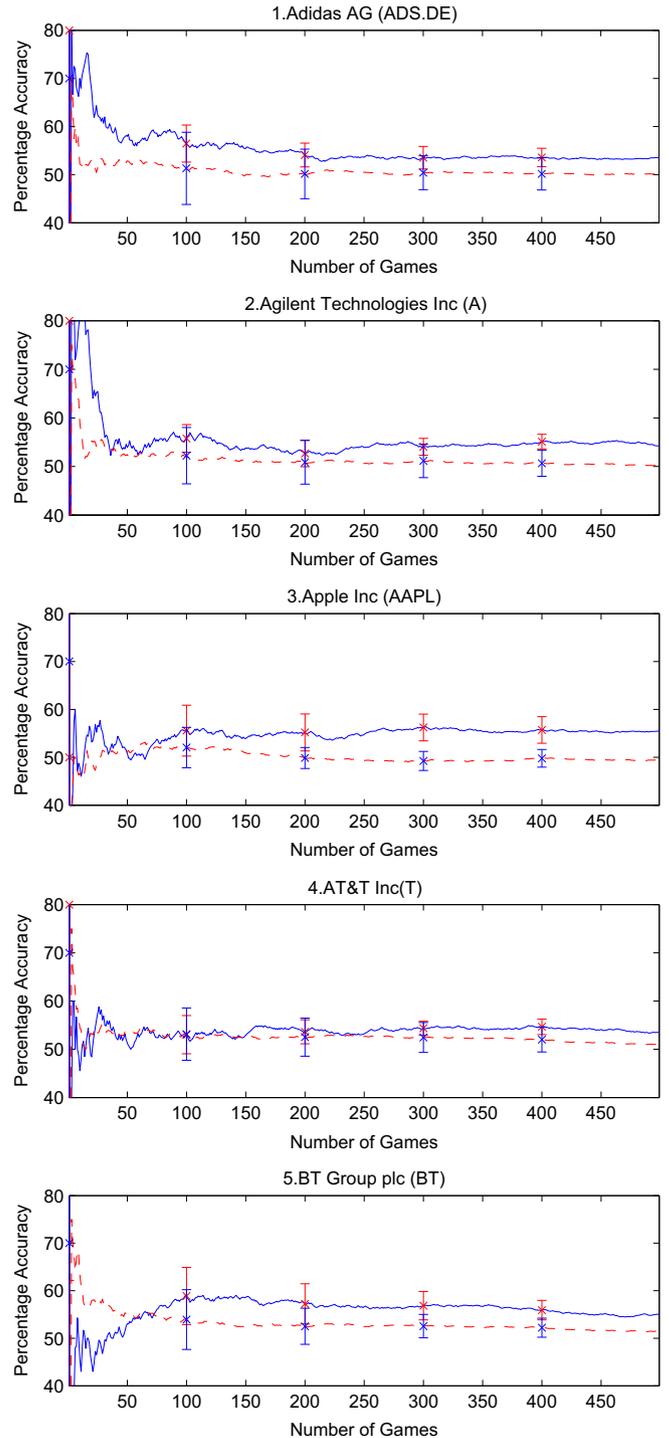


Fig. 7. Experimental results on real-world stock data: averaged prediction accuracy with standard deviation is compared to the random predictor. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

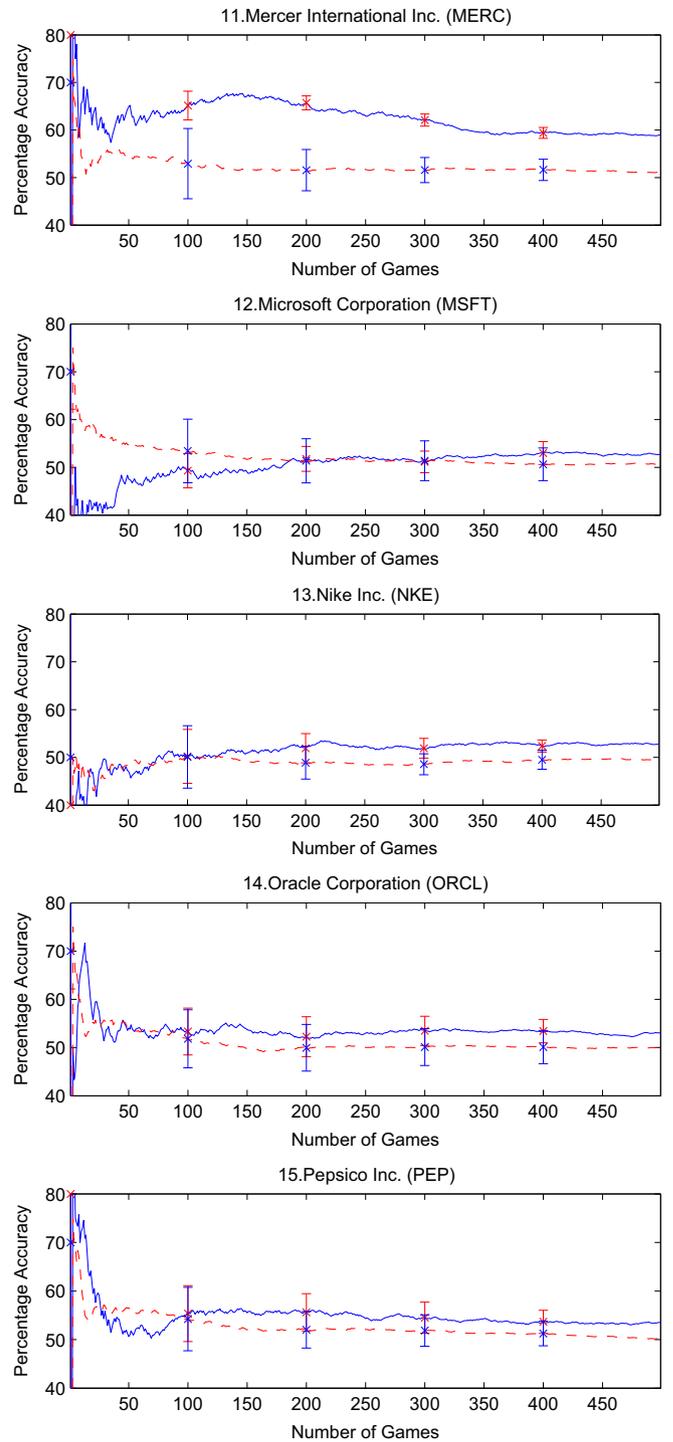
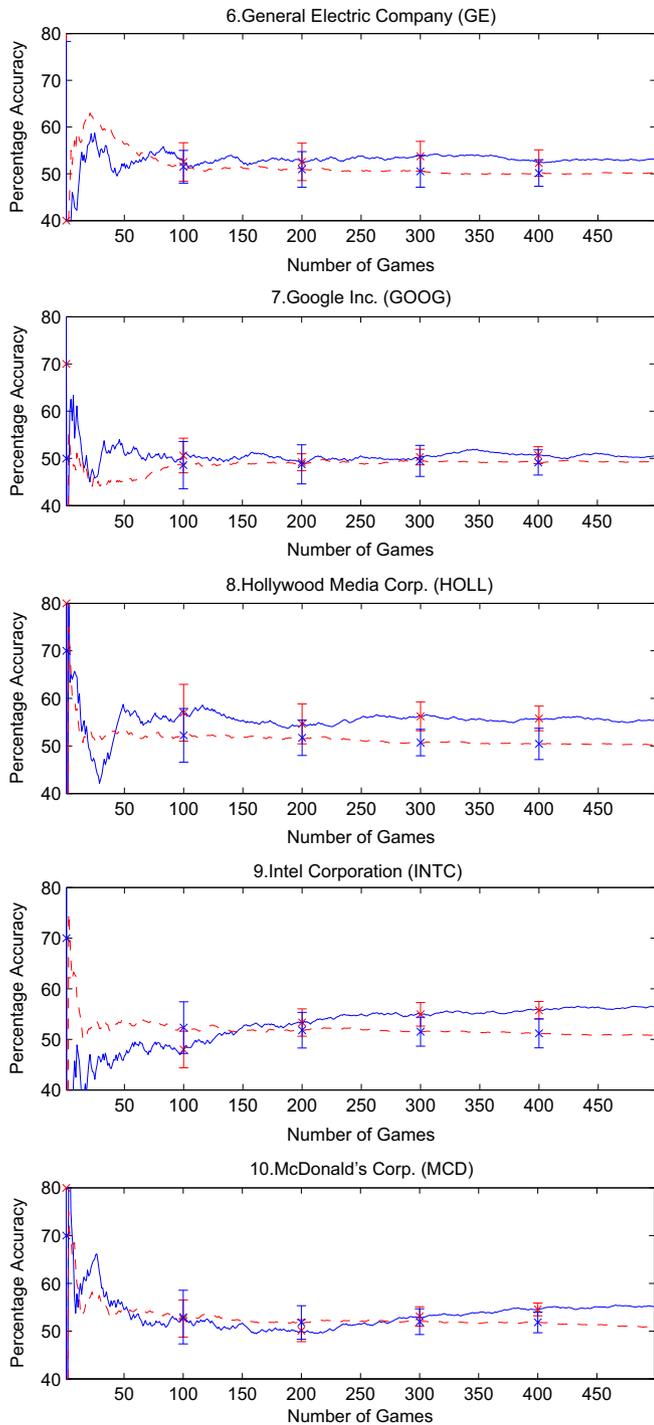


Fig. 8. Experimental results on real-world stock data: averaged prediction accuracy with standard deviation is compared to the random predictor. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

Fig. 9. Experimental results on real-world stock data: averaged prediction accuracy with standard deviation is compared to the random predictor. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

integer) and then use Eq. (14) for normalization:

$$P_N(t) = \frac{P_{max}}{\text{round}(P_o(t))} (P_o(t) - P_c(t)) \quad (14)$$

where $P_N(t)$ is the normalized price differences and they are plotted in Fig. 5. Our aim is that $P_N(t)$ can represent the number of choice 1 ($\#C_1$) within a minority game configuration, it has to be positive integers. We shift $P_N(t)$ and round it to the nearest integer, $\#C_1(t)$ can be obtained as follows:

$$\#C_1(t) = \text{round}(\tau * P_N(t) + \nu) \quad (15)$$

where τ is the scaling factor and ν is the shifting magnitude. The shifting factor is set to $N/2$ in our experiments. Since we have used $N=31$, then $\nu = 15.5$. $\#C_1(t)$ for Nokia Corporation (NOK) dataset is shown in Fig. 6.

5.2.2. Experimental evaluation and analysis

We test our model on 34 real-world stock market data. We took this data from both Chinese and foreign stock markets in different industries so that we can test our model against diverse stock

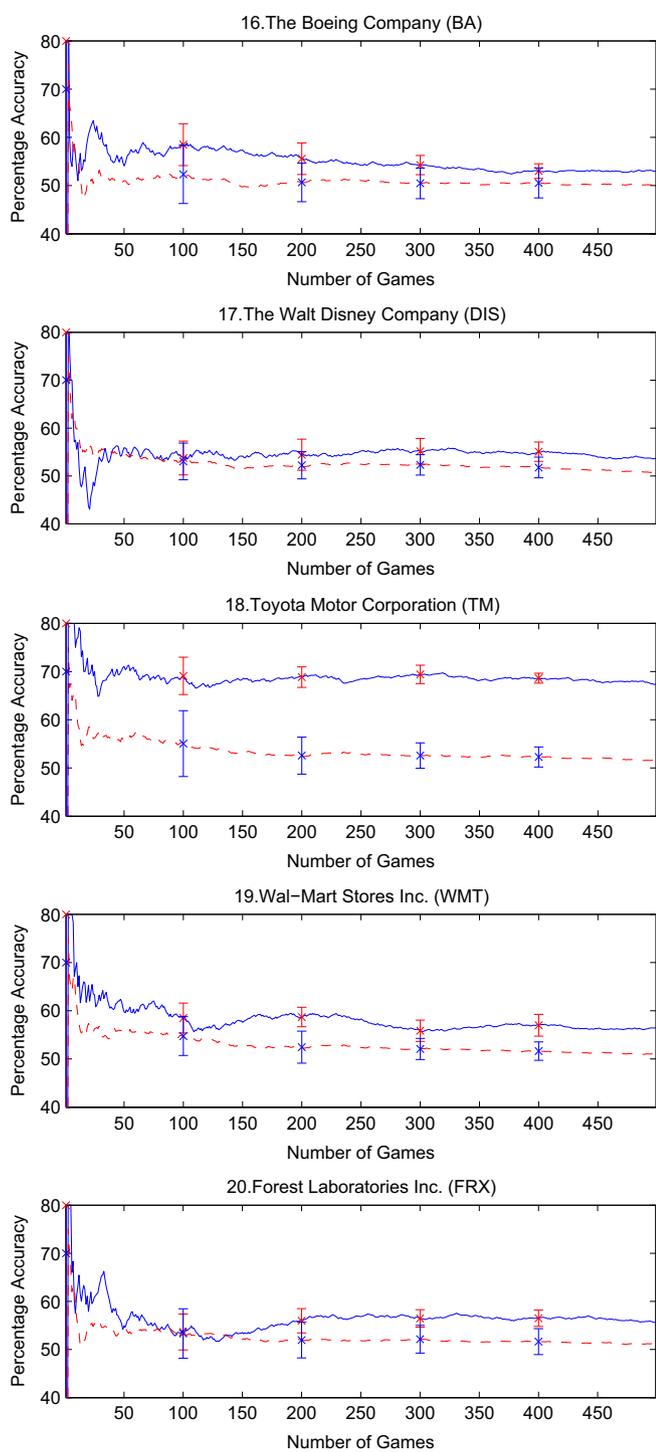


Fig. 10. Experimental results on real-world stock data: averaged prediction accuracy with standard deviation is compared to the random predictor. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

indexes. We take the preprocessed choices ($\#C_1$) data of first 1000 trading days (games) for training and the remaining data for test. All the data is downloadable from Yahoo! Finance.⁵ In order to get stable and unbiased results we conducted the experiment 10 times on each data set and then calculated the corresponding error bars (standard deviations). Table 4 shows the details of the datasets along with the final prediction results. The first column is the

⁵ <http://finance.yahoo.com/>

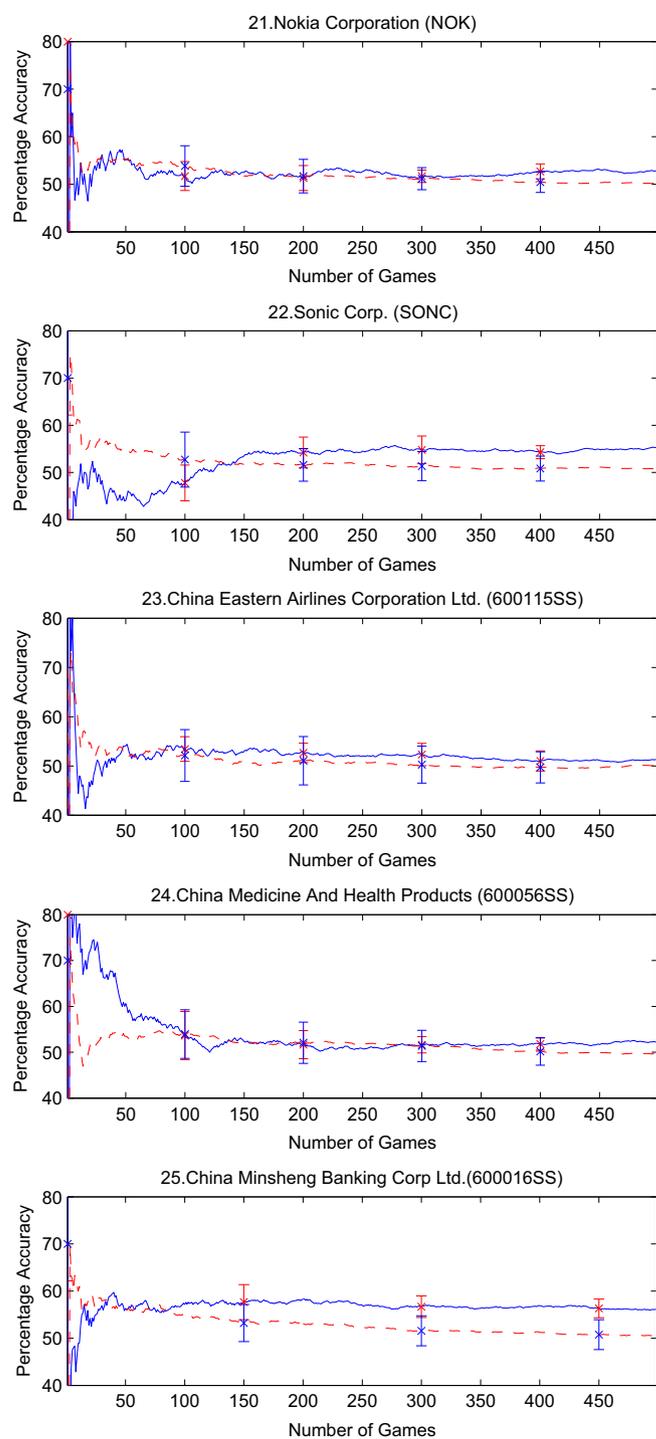


Fig. 11. Experimental results on real-world stock data: averaged prediction accuracy with standard deviation is compared to the random predictor. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

name of the dataset followed by the stock code in parenthesis and the datasets can be searched and downloaded using these codes. The second column shows the starting date of each dataset. The third and the fourth columns show the prediction accuracy after 800 and 1000 games, respectively. The value of normalization parameter τ is shown in the last column.

Figs. 7–13 show the prediction accuracy compared to the random prediction (red-dashed curves) on each data set. Therefore, as long as our prediction accuracy is above the random predictor, we can think that our proposed model has learned the latent

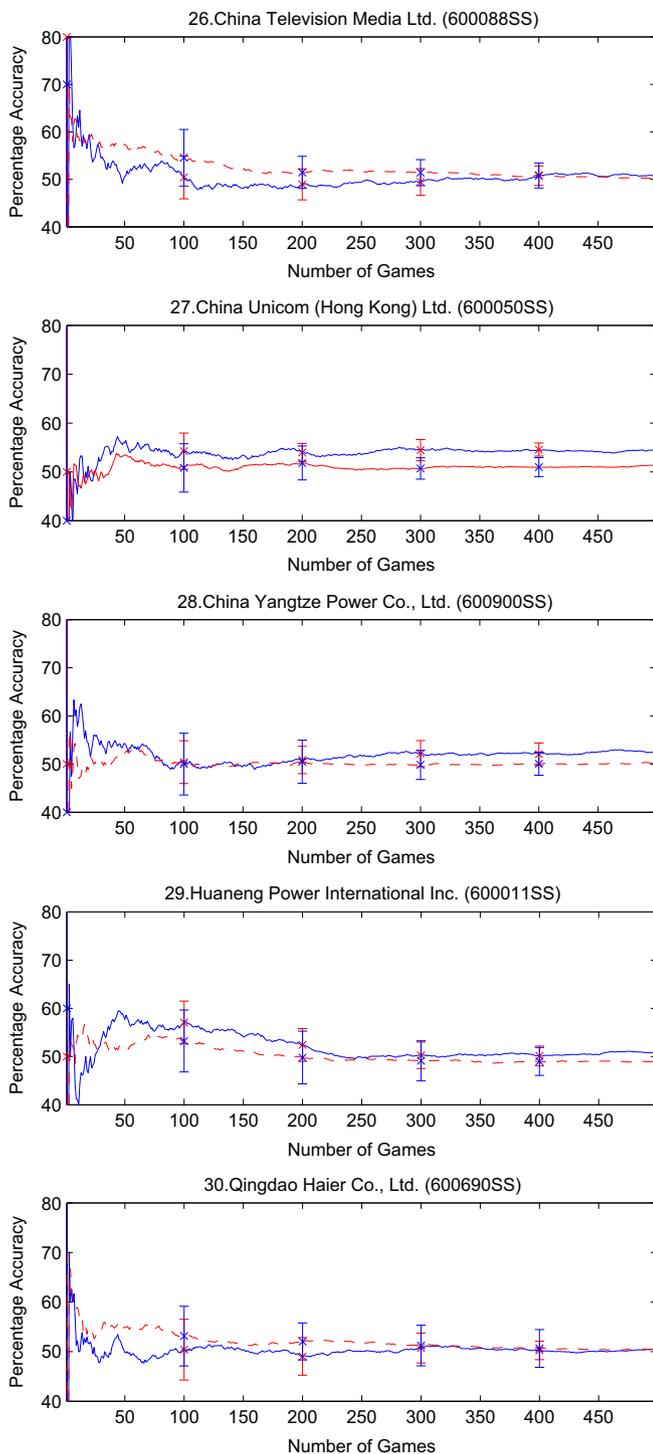


Fig. 12. Experimental results on real-world stock data: averaged prediction accuracy with standard deviation is compared to the random predictor. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

micro–macro relations and can predict the future trends somehow. Given the test data, most of our results are better than random predictions with a few exceptions. Datasets (stocks) 1–6,8,11,13–20,22–23,25,27–29,31,33 perform much better than random prediction. Datasets 7,12,21,32 perform slightly better than the random prediction. Although their accuracy is higher than random prediction, they sometimes overlap with the random prediction. Datasets 9,10 perform less accurately than random prediction for the first 200 games but subsequently their accuracy

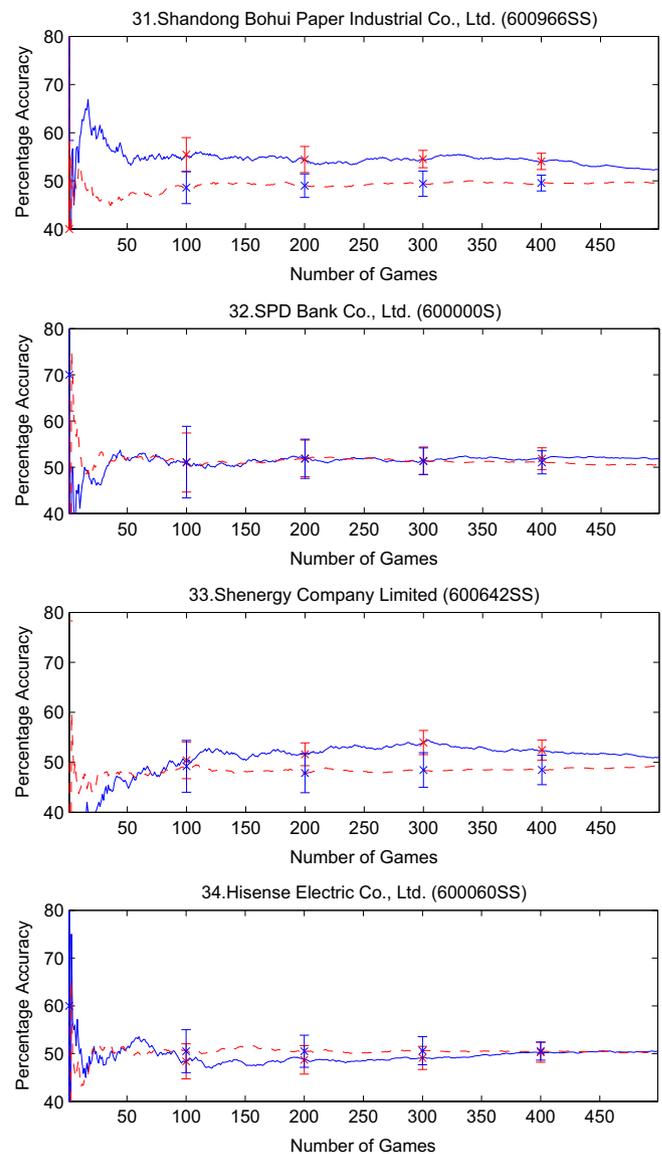


Fig. 13. Experimental results on real-world stock data: averaged prediction accuracy with standard deviation is compared to the random predictor. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

increase and they perform better than random predictions. Datasets 26,30 and 34 perform slightly lower than random prediction but their final accuracy is comparable to random prediction. From Table 4 we can see that the accuracy of our model is between 50.44% and 69% depending on different datasets. These results indicate the presence of statistical patterns within the data can be captured by our model. However, for a minority of test data, our model does not perform that well, but nevertheless it does not go below the baseline accuracy of 50%. The reason for this dip in performance can be either due to the modeling assumptions of our graphical model or the complication of the data itself.

Du et al. [8] proposed a model called Minority Game Data Mining (MGDM), in which the MG is also used to decompose the collective behavior instead of inferring individual behavior from data. Genetic algorithms were used to search for all possible combinations of strategies that yield best results. Table 5 compares the performance of MGDM to our model (GM-MG graphical model based on minority game). Results of our algorithm are at least comparable and in fact outperform the MGDM model on 5 of the 9 datasets.

Table 5

Performance comparisons to the MGDM [8] model.

Dataset name (Index)	Start date	MG-GM	MGDM
China Eastern Airlines (600115SS)	11/05/1997	52.31	55.62
China Minsheng Banking (600016SS)	12/19/2000	56.59	55.71
China Television Media (600088SS)	06/16/1997	49.43	50.69
China Unicom (600050SS)	10/09/2002	54.45	51.34
Huaneng Power (600011SS)	12/06/2001	50.27	50.87
Qingdao Haier (600690SS)	01/02/2001	50.74	49.52
Shandong Bohui Paper (600966SS)	06/08/2004	54.52	53.94
SPD Bank (600000SS)	01/02/2001	51.54	55.99
Shenergy (600642SS)	01/02/2001	53.95	51.78
Hisense Electric (600060SS)	04/22/1997	49.10	53.13

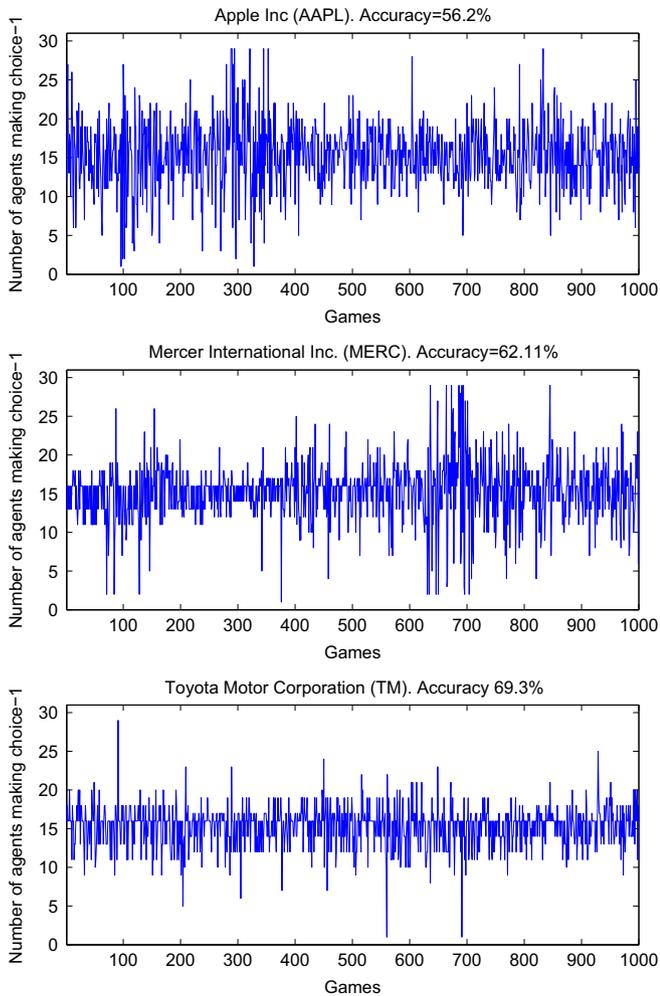
**Fig. 14.** Effects of structure in data on prediction accuracy.

Fig. 14 shows $\#C_1(t)$ for three representational datasets: Apple Inc (AAPL), Mercer International Inc. (MERC) and Toyota Motor Corporation (TM). From this figure we can observe that although the $\#C_1(t)$ for the TM dataset is still apparently random, it has slightly more obvious structure than the MERC dataset. The same can be observed on the MERC and AAPL datasets. As a result it is not surprising that of these three datasets our model has the highest accuracy for the TM dataset and the lowest for the AAPL dataset. In our experiments, we used MS Visual C# 2010 along with Matlab on a 32-bit computer with 3 GB of RAM and two 2.93 GHz processors. For one iteration of Algorithm 1 it takes around 1.1 s if the number of agents N is 31. Therefore the total

duration for 10 iterations of Algorithm 1 for 1000 games is around 1 h and 30 min.

6. Conclusions and Future Work

The collective behavior always seems to be complex, random and unpredictable. Based on the research in evolutionary dynamics, agents following simple games may generate complex collective behavior. The micro–macro relation has been a hot topic in statistical physics. In this paper, we turned our focus on whether such system may have the capability of prediction if we can estimate the agent's behavior by using observable collective data. We proposed a framework of using probabilistic graphical models to investigate micro–macro relations and such a trained model can be used to make future prediction. This can also be viewed as a new machine learning paradigm by decomposing collective behavior to an aggregation of micro-level behavior of agents. We first performed experiments on an artificial data set to validate our model and then test it on the real-world market data. Although finding patterns of real-world market data has always been a controversial topic as it violates the efficient-market hypothesis (EMH) [9]. However, based on our empirical studies, we indeed found statistical significant patterns based on history data. In the future work, we will explore how the factors like memory size, agent size would influence the system. In order to model the complex collective behavior in the real-world, the minority game may not be enough, we need to investigate different games to generate more likely micro-level behavior.

Acknowledgments

This work is supported by the National Science Foundation of China Nos. 61305047 and 61401012.

References

- [1] W.B. Arthur, Bounded rationality and inductive behavior (the El Farol problem), *Am. Econ. Rev.* 84 (1994) 406–411.
- [2] C.M. Bishop, *Pattern Recognition and Machine Learning*, Springer-Verlag, New York, 2006.
- [3] A. Chakraborti, D. Challet, A. Chatterjee, M. Marsili, Y.-C. Zhang, B. K. Chakrabarti, Statistical mechanics of competitive resource allocation using agent-based models, *Phys. Rep.* 552 (2015) 1–25.
- [4] D. Challet, Y.-C. Zhang, Emergence of cooperation and organization in an evolutionary game, *Physica A* 246 (1997) 407–418.
- [5] D. Challet, M. Marsili, R. Zecchina, Statistical mechanics of systems with heterogeneous agents: minority games, *Phys. Rev. Lett.* 84 (2000) 1824–1827.
- [6] D. Challet, M. Marsili, Y.-C. Zhang, Stylized facts of financial markets and market crashes in minority games, *Physica A* 294 (2001) 514–524.
- [7] D. Challet, M. Marsili, Y.-C. Zhang, *Minority Games: Interacting Agents in Financial Markets*, Oxford University Press, USA, 2004.
- [8] Y. Du, Y. Dong, Z. Qin, T. Wan, Exploring market behaviors with evolutionary mixed-game learning model, In: *Proceedings of the International Conference on Computational Collective Intelligence, ICCCI'11, Lecture Notes in Computer Science*, vol. 6922, Springer-Verlag, Berlin, Heidelberg, 2011, pp. 244–253.
- [9] E. Fama, Efficient capital markets: a review of theory and empirical work, *J. Finance* 25 (1970) 383–417.
- [10] J.D. Farmer, D. Foley, The economy needs agent-based modelling, *Nature* 460 (August) (2009) 685–686.
- [11] D. Gode, S. Sunder, Allocative efficiency of markets with zero-intelligence traders: market as a partial substitute for individual rationality, *J. Political Econ.* 101 (1) (1993) 119–137.
- [12] N.F. Johnson, P. Jeffries, P.M. Hui, *Financial Market Complexity*, Oxford University Press, Oxford, 2003.
- [13] B.J. Kim, A. Trusina, P. Holme, P. Minnhagen, J.S. Chuang, M.Y. Choi, Dynamic instabilities induced by asymmetric influence: prisoner's dilemma game in small-world networks, *Phys. Rev. E* 66 (2002) 021907.
- [14] D. Koller, N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*, MIT Press, Cambridge, 2009.

- [15] G. Li, Y. Ma, Y. Dong, Z. Qin. Behavior learning in minority games. In Collaborative Agents—Research and Development, Lecture Notes in Computer Science, vol. 6066, Springer-Verlag, Berlin, Heidelberg, 2011, pp. 125–136.
- [16] X. Li, Y. Wu, Z. Rong, Z. Zhang, S. Zhou, The prisoner's dilemma in structured scale-free networks, *J. Phys. A: Math. Theory* 42 (2009) 245002.
- [17] T.S. Lo, P.M. Hui, N.F. Johnson, Theory of the evolutionary minority game, *Phys. Rev. E* 62 (September) (2000) 4393–4396.
- [18] Y. Ma, G. Li, Y. Dong, Z. Qin. Minority game data mining for stock market predictions, in: AAMAS Workshop on Agents and Data Mining Interaction, ADMI'10, Lecture Notes in Computer Science, vol. 5980, Springer-Verlag, Berlin, Heidelberg, 2010, pp. 178–189.
- [19] T. Minka, J.M. Winn, J.P. Guiver, D.A. Knowles, Infer.NET 2.4, 2010. Microsoft Research Cambridge (<http://research.microsoft.com/infernet>).
- [20] M. Nowak, *Evolutionary Dynamics*, Harvard University Press, Cambridge, MA, 2006.
- [21] M. Nowak, Five rules for the evolution of cooperation, *Science* 314 (December) (2006) 1560–1563.
- [22] M. Perc, P. Grigolini, Collective behavior and evolutionary games c an introduction, *Chaos Solitons Fractals* 56 (November) (2013) 1–5.
- [23] M. Perc, A. Szolnoki, Coevolutionary games—a mini review, *BioSystems* 99 (2010) 109–125.
- [24] Z. Qin. Market mechanism designs with heterogeneous trading agents, in: Proceedings of Fifth International Conference on Machine Learning and Applications (ICMLA-2006), 2006, pp. 69–74.
- [25] Z. Qin, Y. Dong, and T. Wan. Evolutionary models for agent-based complex behavior modeling, In: Artificial Intelligence, Evolutionary Computing and Metaheuristics, *Studies in Computational Intelligence*, 2013, pp. 601–631.
- [26] Z. Qin, T. Wan, Y. Dong, Y. Du, Evolutionary collective behavior decomposition model for time series data mining, *Appl. Soft Comput.* 26 (2015) 368–377.
- [27] A. Rapoport, A. Chammah, C. Orwant, *Prisoner's Dilemma: A Study in Conflict and Cooperation*, University of Michigan Press, Ann Arbor, Michigan, 1965.
- [28] V.L. Smith, An experimental study of competitive market behavior, *J. Political Econ.* 70 (April (2)) (1962) 111–137.
- [29] A. Szolnoki, Z. Wang, M. Perc, Wisdom of groups promotes cooperation in evolutionary social dilemmas, *Sci. Rep.* 2 (2012) 576.
- [30] M. Tomochi, Defector's niches: prisoner's dilemma game on ordered networks, *Networks* 26 (2004) 309–321.
- [31] J. Winn, C.M. Bishop, Variational message passing, *J. Mach. Learn. Res.* 6 (2005) 661–694.

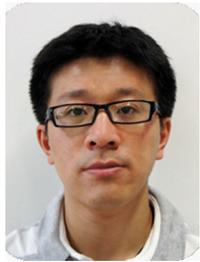


Farhan Khawar was formerly with the Intelligent Computing and Machine Learning Lab at Beihang University and is currently a PhD student at The Hong Kong University of Science and Technology. His research interests mainly focus on Machine Learning and Probabilistic Graphical Models.



Tao Wan is an assistant professor in School of Biological Science and Medical Engineering. She was a research associate at the Case Western Reserve University and Boston University. She received her Master degree in Global Computing and Multimedia from the University of Bristol, UK in 2004 and her PhD in Computer Science from the same university in 2009. She spent one year working as a senior researcher in the Samsung Advanced Institute of Technology (SAIT) China before becoming a visiting scholar in the Visualization and Image Analysis Lab in the Robotics Institute, Carnegie Mellon University. Her research interests are statistical models for image segmentation, fusion,

and denoising, machine learning, computer-aided diagnosis system, medical image analysis on prostate and breast cancer.



Zengchang Qin obtained his MSc in Computer Science and PhD in Artificial Intelligence from the University of Bristol, UK, in 2002 and 2005, respectively. He worked as a lecturer in the same university before joining Lotfi Zadeh's BISC group at the EECS Department of UC Berkeley as the BT postdoctoral fellow in 2006. He has been working in Beihang University as an associate professor in the School of Automation Science and Electrical Engineering from 2009. He was also a visiting scholar at Robotics Institute, Carnegie Mellon University, from November 2010 to June 2011. His research interests are uncertainty modeling, machine learning, multimedia retrieval and agent-based modeling.