Research Article

A Study of Prisoner's Dilemma Game Model with Incomplete Information

Xiuqin Deng¹ and Jiadi Deng²

¹*School of Applied Mathematics, Guangdong University of Technology, Guangzhou, Guangdong 510006, China*

²*Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China*

Correspondence should be addressed to Xiuqin Deng; xiuqindeng@163.com

Received 25 May 2014; Revised 22 September 2014; Accepted 23 September 2014

Academic Editor: Yiu-ming Cheung

Copyright © 2015 X. Deng and J. Deng. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Prisoners' dilemma is a typical game theory issue. In our study, it is regarded as an incomplete information game with unpublicized game strategies. We solve our problem by establishing a machine learning model using Bayes formula. The model established is referred to as the Bayes model. Based on the Bayesian model, we can make the prediction of players' choices to better complete the unknown information in the game. And we suggest the hash table to make improvement in space and time complexity. We build a game system with several types of game strategy for testing. In double- or multiplayer games, the Bayes model is more superior to other strategy models; the total income using Bayes model is higher than that of other models. Moreover, from the result of the games on the natural model with Bayes model, as well as the natural model with TFT model, it is found that Bayes model accrued more benefits than TFT model on average. This demonstrates that the Bayes model introduced in this study is feasible and effective. Therefore, it provides a novel method of solving incomplete information game problem.

1. Introduction

Incomplete information games are influenced by the private information owned by at least one game player, such as current game state, mechanism of other players in decision-making, the game state of other players, and the reward/punishment mechanism of the game [1]. However, due to the absence of its optimal or relatively optimal solution (at any certain game state), such an incomplete information game is insoluble by traditional study methods. This is due to the fact that the other players restrain the strategy of incomplete information games. Harsanyi [2] analyzed an incomplete information game using Bayesian game player strategies and proposed methods for modeling and analyzing game problems. Zinkevich et al. [3] investigated incomplete information games using Nash equilibrium and minimizing regret method and proposed some game strategies for improving poker game problems.

As a branch of artificial intelligence and a cutting-edge research topic, machine learning has been paid great attention in related fields in recent years. Machine learning is defined as a research method aiming at obtaining a

more desired approximate solution based on the general rules obtained by analyzing a large amount of data [4]. Statistical machine learning is a branch of machine learning. It integrates statistical theory into machine learning by combining probability theory and stochastic mathematical knowledge with machine learning to improve the efficiency and accuracy [5, 6]. The Bayesian classification algorithm is a commonly used machine learning method [7]. The simplified model of naive Bayesian classification is often used in text classification.

The prisoners' dilemma game is a classic cooperation and selection problem based on the assumption of selfish human motives [8]. It is popular and widely applied in mathematics and economics [9]. For a long time, it has been a classical game theory problem and attracted great interest from mathematics and economics researchers around the world. Game theory was born in the mid-twentieth Century and was founded by von Neumann (a famous mathematician and founding father of computing) and Morgenstern (a famous economist). The starting point for the development of game theory was the publication of John von Neumann and Oscar Morgenstern's seminal work *The Theory of Games and*

Economic Behavior in 1944 [10]. Game theory brought radical changes to economics and provided a standard analysis tool for economists. In light of the contributions of game theory to economics, the Royal Swedish Academy of Sciences awarded Nobel Prizes for economics to Nash, Harsanyi, and Selten in 1994 and Aumann and Schelling in 2005, respectively [10]. In the famous artificial intelligence algorithm competition of prisoners' dilemma, Axelrod concluded that a TFT (tit-for-tat) model was the optimum solution through brute force competition in participating algorithms [11, 12]. Miller [13] introduced an automaton model to simplify and analyze prisoners' dilemma and proposed a more general prisoners' dilemma decision analysis method. He also applied the model to solve problems arising from generalisation. And Press suggested that the prisoner's dilemma is an ultimatum game and gave an example of strategy, which can gain an unfair share of rewards, to support his claim [14]. Using genetic algorithm, Lin and Wu [15] studied the evolution of strategies in the iterated prisoner's dilemma on complex networks and found that the agents located on complex networks can naturally develop some self-organization mechanics of cooperation, which can not only result in the emergence of cooperation but also strengthen and sustain the persistent cooperation.

Evolutionary game theory [16, 17] extends and combines ideas from game theory and evolutionary biology to study the evolution of an interacting population of individuals. Perhaps one of the simplest games in evolutionary game theory is the so-called evolutionary spatial prisoners' dilemma (ESPD) [18]. Cardillo et al. [19] investigated the coevolution of strategies and update rules in the evolutionary spatial prisoners' dilemma (ESPD). The authors concluded, for a variety of underlying graph topologies, that when the dynamics coevolves with the strategies it leads to more cooperation in the weak prisoners' dilemma in general. Du et al. discussed another evolutionary method that uses the improved weighted network to solve the problem [20]. Literature [21] proposed a model using two graphs in conjunction with the ESPD: one for determining player interaction and second for updating strategies. Moreover, Wang et al. have proposed some evolutionary algorithm to solve relevant problems [22, 23]. Game theory techniques have been widely applied to various engineering design problems in which the action of one component has impact on (and perhaps conflicts with) that of any other component.

Prisoners' dilemma could be regarded as a game with incomplete information. It satisfies the conditions for an incomplete information game; namely, the players of each game are incapable of determining the choice of their rival in any current station. In our study, we propose a naive Bayesian classification method, which is used to establish the machine learning model for prisoners' dilemma in an attempt to solve it through statistical machine learning. With the use of Bayesian classification, the opponents' strategy can be presented as the possibility of choice which means the accuracy of the prediction on opponents' strategy has been promoted. Moreover, we introduce an evaluation with multiple processes to provide the information with high precision for the final decision in our strategy. In the step of record, we suggest some efficient data structures and ensure

FIGURE 1: The incomes of the models after 10,000 games.

a reasonable space and time complexity in our method. We test the proposed method during the competitions with some typical methods. The simulation experimental results show that our method outperforms four classical methods (see Figure 1 for more details). We further apply the Bayes method to multiplayer games. The simulation result indicates that the Bayes method gains the highest income in the multiplayer test (see Figure 5 for more details).

2. Game Model

2.1. Game with Incomplete Information. Games with incomplete information can be defined as the game where the players are uncertain about some important parameters [2]. Generally, the incomplete information can be regarded as players' lack of full information on some basic law of the game. Such incomplete information can mainly arise in three different ways as follows.

- (1) Lack the information of the profit function of the game: the profit function is a function with N or more parameters to get the final income of each player's profit (N means the total number of players). It can be represented as

$$p = P(s_{x1}, s_{x2}, \dots, s_{xN}, p_1, p_2, \dots, p_m), \quad (1)$$

where $s_{x1}, s_{x2}, \dots, s_{xN}$ is all the N players' strategy set and $p_1, p_2, \dots, p_m$ is the other parameters involving the outcome, like the number of turn.

- (2) Lack the information of the strategy of other players in the game; a strategy list can be defined as the decision of a player made in each step (L is the temporary number of turn):

$$s_i = \{d_1, d_2, \dots, d_L\}. \quad (2)$$

Players will make their decision for next step based on the previous situation of the game

$$d_{l+1} = D(\{g_1, g_2, \dots, g_l\}). \quad (3)$$

- (3) Lack the information of the available choice space of others in the game: each player's choice may be

limited by the rule or background of the game. That is to say, the players' action in each step is finite but uncertain.

We can say that all other cases of incomplete information can be reduced into these three basic cases [1]. In the prisoner's dilemma, the player is not aware of the other player's strategy (case 2) for there is no best or winning strategy in the game. Yet the profit function (the outcome according to each player's choice) and the choice space (cooperate or betray) are definite in the game.

In our research, we mainly focus on the game in case 2 where players know nothing about others' strategy. Such game model can be found in various ranges of areas. For example, in some economical model, the choice space is limited by the situation of the problem and the profit curves can be found from the previous models. And the strategy of each competitor will be unpredictable in the case that the profit is not linear or varied by time (as it is in reality). So such model can be regarded as a game with incomplete information in case 2.

2.2. Game of Prisoners' Dilemma. In this game, two individuals determine cooperation or defection. If the two individuals are mutually cooperative, they both earn incomes R ; if they defect to each other, the incomes of both sides are P ; if one individual is cooperative while the other is in a state of betrayal, the cooperative one gains S , while the treacherous player gains T (Table 1). Here, $T > R > P > S$, and $2R > S + T$. The latter formula means that total income of the two cooperative individuals is always larger than that gained in case of one individual's treachery. However, with regard to individuals, the incomes earned by defection to cooperation are greater than that by cooperation to cooperation.

In this experiment, the parameters selected were consistent with those that Axelrod and Hamilton [12] used in solving the prisoners' dilemma. That is to say, the incomes were $R = 3$, $T = 5$, $S = 0$, and $P = 1$, which satisfied the conditions: $T > R > P > S$ and $2R > S + T$.

In addition, each pair of strategies was competed for L_m times; namely, both sides had to make L_m selections. The result of each turn was recorded and the information of opponents' selection was sent to each player in the end of each turn. After L_m turns, the results of each pair of strategies were listed in forms of their total score. According to the total score, the strategy of corresponding strategy models was evaluated. In general competition, L_m was set as 10,000. It can be seen that the game, in the long-term, could yield stable income results.

Several typical strategy models are presented in the following.

- (1) TFT (tit-for-tat) strategy, that is, "return like-for-like" strategy: TFT is a well-known model for prisoners' dilemma. The main idea of this strategy is that, by starting with cooperation, the strategy selection of a round is made on the basis of the selection from the previous round. That is, if the rival selects cooperation or defection in the previous round, the selection will be repeated in the current round. This

TABLE 1: Game information.

(Player A, Player B)	Cooperation (C)	Defection (D)
Cooperation (C)	(R, R)	(S, T)
Defection (D)	(T, S)	(P, P)

strategy performed best in the artificial intelligence algorithm competition organized by Axelrod and Hamilton [12] (although the TFT strategy in this study was consistent in concept with that TFT strategy, the difference here lies in its details).

- (2) PTFT, an improved TFT strategy [24]: this strategy is relatively more selfish than TFT. It still starts with cooperation; however, in the following rounds, cooperation is only selectable in the case of an absence of defection for three rounds.
- (3) GTFT, another improved TFT strategy [8]: its strategy allows a certain probability of cooperation in the case of rival defection and a certain probability of defection in the case of cooperation. It solves the deadlock arising from mutual defection in the competition.
- (4) Pavlov, a different strategic concept [8]: it bullies the weak and fears the strong; namely, cooperation is continued in cases of mutual cooperation. However, defection is selected when one side chooses defection. Moreover, in the case of mutual defection, cooperation is given priority. Such a strategy represents a local optimum in genetic algorithm terms.
- (5) Random, random strategy, that is, randomly returning to cooperation or noncooperation: in related programs, a 50 : 50 random strategy is more commonly applied; that is, the probability of returning to cooperation or defection is 50%. This strategy is mainly adopted to assess fixed strategies, set competition parameters, and so forth.
- (6) Normal, a strategy mode developed by simulating common players: in this strategy, cooperation or defection would be selected with different probabilities based on the selections of both sides in the previous game. This strategy simulates player participation in a game using different strategies.

2.3. Extended Prisoners' Dilemma's Model. The original prisoners' dilemma problem contains only two players with two different choices. And the profit function of the game is given and fixed during the game. However, most of the problems in reality are not in such simple scene but multiplayer variants. Therefore, multiplayer formats afford the opportunity for a useful expansion of the prisoners' dilemma game.

In the extended prisoners' dilemma, N players will be considered in a game. A time-invariant profit function is given. The profit function has N parameters and for each set of input only one corresponding output would be produced. The choice set contains two elements, the cooperation or the defection. That is to say, each player can only select one of these two choices as their action. The game would have L_m

rounds. In one round, all players have to give their choice to the judges at the same time and then get the feedback. The feedback includes other players' choices. As the profit function is provided for the game, all players can get the competitors' profit through the feedback which includes all players' choice. As a game with incomplete information, all the players could mask their strategy. Plus, other players' choice would be unknown before the step of returning feedback.

By referring to a published multiplayer dilemma study [25], a reward and punishment rule was defined for this study as follows.

- (1) When all players selected cooperation (C), their income R was averaged across each player.
- (2) When partial players selected defection (D), their income T was averaged out among the treacherous players, while income S was shared amongst cooperatives players.
- (3) When all players selected defection (D), the income P was averaged out among all players.

For a prisoners' dilemma with $n = 4$, the parameters were set to $R = 12$, $T = 10$, $S = 0$, and $P = 4$, which was in agreement with the standard form of the game. That is, the individual optimum solution of every player was obtained when one player selected defection, while the overall optimum solution was obtained when all players selected cooperation.

In this prisoners' game, the four strategies of each group were unavailable to the other players before they made their decision. After decisions were made, the income of each player was calculated and the decisions were revealed: the game was repeated 10,000 times.

3. Strategy in Prisoner's Dilemma Game

In the prisoner's dilemma game we are studying, the strategy of other players is unknown while the profit function and choice space are clear. Our strategy in this game is to reduce the unclear information and maximize the profit in expectation. There are at least three challenges as follows.

- (i) The strategies of other players are variable and unstable. They may make a different choice in the same situation (like the strategy to randomly make a choice). That is to say, no best choice can be selected in a single game.
- (ii) All of the other players in the game will have different strategy. And they play together in one game.
- (iii) The performance and efficiency of the strategy should be promised, especially in the case that the number of players is large.

To solve the problem listed above, we propose the Bayes formula is the basic method of our strategy. Though players choice can be considered being random and irregular, their action can be described as a serial of probabilities of the possible choices. The Bayes formula provides us with a way to

TABLE 2: Decision possibility table of TFT.

Choice of last turn (my opponent's)	My choice of next turn (C)	My choice of next turn (D)
(C, C)	1.0	0
(C, D)	0	1.0
(D, C)	1.0	0
(D, D)	0	1.0

make the prediction based on the history. According to Bayes formula, we can build our prediction table, which includes the probability sets to individual player. As the game in our study is a multiple-turn-based game, the historical data is easy to get and restore.

After the prediction, we could evaluate each choice in our choice space with some probability-based method, for instance, the expectation of profit. To be more convincing, we can include the multiple future steps in our evaluation. Finally, we select the one choice with the highest value as the decision of temporary turn.

To make our strategy more effective, we can use n -dimension array or hash table for data storage. N -dimension array shows best performance in the competition with small number of players. Hash table is used in game including a large number of players.

3.1. Prediction. In an extended prisoner's dilemma game's model, we assume that there are L_m players in the game and each player has K choices. The goal of prediction is to get the possibility distribution of each different players' choice. That is,

$$P(X_i = c_k). \quad (4)$$

We do not know each different players' strategy, but we can get each player's choice in the past. Our strategy assumes that other players would base their decision on the historical data they recorded and the decision possibility table. Other players would record limited steps of historical data and use a decision possibility table to make their decision. For example, Table 2 shows the decision possibility table, which is used in TFT. As a consequent, we can infer such table with the use of some tools from probability theory including Bayes formula and make a prediction of other players' choice based on the history data.

The decision possibility table can be presented as the possibility function $P(X_i = c_k | H = h_t)$, which means the possibility of player i (X_i) to choose c_k when the history of opponents' choice is h_t . From the Bayes formula [26], we can know that

$$P(X_i = c_k | H = h_t) = \frac{(P(X_i = c_k) \prod_j P(X_{j,m} = h_{j,t} | X_i = c_k))}{(\sum_k (P(X_i = c_k) \prod_j P(X_{j,m} = h_{j,t} | X_i = c_k)))}, \quad (5)$$

where X_i means the choice of player i , c_k presents the k th choice, H represents the history of the choice of all players,

and h_t is an m -dimension vector meaning the temporary record of the players' decisions

$$\begin{aligned} h_j &= \{R_{j-1}, R_{j-2}, \dots, R_{j-m}\} \\ &= \left\{ \{r_{1,j}, r_{2,j}, \dots, r_{N,j}\}, \dots, \{r_{1,j-m}, r_{2,j-m}, \dots, r_{N,j-m}\} \right\}, \\ &\quad (r_{i,j} \in \{c_0, c_2, \dots, c_k\}), \\ h_{i,l} &= \{r_{i,l-1}, r_{i,l-2}, \dots, r_{i,l-m}\}, \end{aligned} \quad (6)$$

where R_j is a set of choices of each player in the j th turn. And the record set h_j dates back to m turns of the records and includes m set of records and $r_{i,j}$ is the player i th choice in the j th turn, which is one of the elements in the set of choices. $X_{j,m}$ is the m -step choice history of player j . Here, we consider that all the strategies would base their decision on the last few steps and therefore the historical choices would be immaterial at the temporary judge. If the m is very large or in some extreme cases, there may not be enough records for building h_j . The denominator of the formula may become 0. To avoid such situation, we can make a correction for the original formula by simultaneously increasing the molecule and denominator

$$\begin{aligned} P(X_i = c_k | H = h_t) \\ = \frac{(P(X_i = c_k) \prod_j P(X_{j,m} = h_{j,t} | X_i = c_k)) + 1}{(\sum_k (P(X_i = c_k) \prod_j P(X_{j,m} = h_{j,t} | X_i = c_k))) + |c_k|}. \end{aligned} \quad (7)$$

Next step is to get the historical record for the possibility of each situation. From the previous analysis, we know that we should get the extra value of $P(X_i = c_k)$ and $P(H = h_j | X_i = c_k)$ from $\{R_j\}$. Basically, we can easily get the probability from the formula that

$$\begin{aligned} P(X_i = c_k) &= \frac{\sum_L \tau_k(r_{i,l})}{\sum_L \sigma_i(r_{i,l})}, \\ \prod_j P(X_{j,m} = h_{j,t} | X_i = c_k) \\ &= P(X_{1,m} = h_{1,t}, X_{2,m} = h_{2,t}, \dots, X_{N,m} = h_{N,t} | X_i = c_k) \\ &= \frac{\sum_{l=m}^L \prod_{j=1}^N \tau_{i,k,t}^*(h_{j,l})}{\sum_{l=m}^L \prod_{j=1}^N \sigma_t^*(h_{j,l})}, \\ \tau_k(r_{i,l}) &= \begin{cases} 1, & (r_{i,l} = c_k, r_{i,l} \in R_l), \\ 0, & (\text{else}), \end{cases} \\ \sigma_i(r_{i,l}) &= \begin{cases} 1, & (r_{i,l} \in R_l), \\ 0, & (\text{else}), \end{cases} \\ \tau_{i,k,t}^*(h_{j,l}) &= \begin{cases} 1, & (r_{i,l} = c_k, h_{j,l} = h_{j,t}), \\ 0, & (\text{else}), \end{cases} \\ \sigma_t^*(h_{j,l}) &= \begin{cases} 1, & (h_{j,l} = h_{j,t}), \\ 0, & (\text{else}). \end{cases} \end{aligned} \quad (8)$$

Here, $h_{i,l}$ is an ordered and comparable sequence of player i th m -step records and t is the temporary number of turn. We can get the result of formula (7) from the data we record.

Moreover, as the strategy varies from player to player, we should not expect all players to use similar strategies in the game. For example, some players may ignore the choices made by their own and focus on others' choice. The records from the player himself should be abandoned when making the prediction. So, we have to add a weight for each history record. The weight function will be relevant to the player and the choice:

$$\omega_i = w_i(X_j, c_k), \quad (9)$$

and the probability formula would be

$$\begin{aligned} \prod_j P(X_{j,m} = h_{j,t} | X_i = c_k) \\ = \frac{\sum_{l=m}^L \prod_{j=1}^N w_i(X_j, c_k) \tau_{i,k,t}^*(h_{j,l})}{\sum_{l=m}^L \prod_{j=1}^N w_i(X_j, c_k) \sigma^*(h_{j,l})}. \end{aligned} \quad (10)$$

The weight function, which distributes the weight to each history record, would vary in different strategies. In the experiment of depth of considered step of weight functions, we could find that the depth within 5 could preform similarly in the game (see Figure 13). So in our study, we use an one-turn weight function for prediction. Plus, we can find that most of the classical strategies ignore the self-made choices and regard all the choices to be the same. Therefore, we can get one of the weight functions by concluding these attributes from classical strategies:

$$w_i(X_j, c_k) = \begin{cases} 1, & (i \neq j), \\ 0, & (\text{else}). \end{cases} \quad (11)$$

3.2. Income Evaluation. In this step, we must evaluate each choice we can return and choose one of them as the final decision. We would select the choice with the highest score after the evaluation

$$C_t = \arg \max_k \text{val}(c_k, h_t). \quad (12)$$

We can judge the evaluation based on the possibility of each player's choices and the profit function. As the situation that the profit function is given, we can easily predict the value that our player would make from the choice. In one-step prediction, the value equals the expectation of the income. Consider

$$\begin{aligned} \text{val}(c_k, h_t) \\ = \sum \text{profit}(c_k, c_{x1}, \dots, c_{xN}) \\ * P(X_1 = c_{x1}, \dots, X_N = c_{xN} | H = h_t). \end{aligned} \quad (13)$$

The function "profit" is a given profit function with N parameters, which is equal to the number of players; the return will be a vector of the income of the players who make the same choice as the first parameter. We can make our value

```

Function  $p\_step(c_k, p)$ {
  if  $p < 0$  then
    return 0
   $v \leftarrow 0$ 
  for each choice  $c_k$  do
     $v \leftarrow v + \text{val}(c_k, h_t)$ 
     $\text{Append}(\text{ht}, \{c_k, c_{x_2}, \dots, c_{x_N}\})$ 
    for each choice  $c$  do
       $v \leftarrow v + p\_step(c, p - 1)$ 
    end
     $\text{Remove}(\text{ht}, \{c_k, c_{x_2}, \dots, c_{x_N}\})$ 
  end
  return  $v$ 
}

```

ALGORITHM 1

function more visionary if both the N and the K are small (like the classical prisoner's dilemma where $N = 2, K = 2$). A p -step prediction of the income will be a more efficient method. We can get the best choice by the recursive program (see Algorithm 1).

3.3. Feedback Record. At the end of each turn of the game, we can get the feedback from the system. The feedback includes each player's choice and the income they get. In our model, the profit function is given, so we can detect the income of all players with the historical records of all players.

The content of feedback can be represented as follows:

$$F_l = \{c_{x1}, c_{x2}, \dots, c_{xN}\} \quad (c_{xi} \in \{c_1, c_2, \dots, c_K\}). \quad (14)$$

We can record the feedbacks as a list. As a list, the space complexity of the record is $O(LN)$. In the step of prediction, the time complexity is $O(KN^2L^2)$. In the step of income evaluation, the time complexity is $O(NK^N)$. In the step of record, the time complexity is $O(N)$. In our discussion, the number of players and the number of the choices is relatively small and the number of turns is large. Such that $N, K \ll L$. And we can find that the bottleneck of the problem will be the time complexity of prediction.

To improve the problem that all the records have the same weight and consider one historical step, we can use a N -dimension array for storage. We can build an N -dimension array A where each dimension's length is K . The entries of A are counters of all specific situations, which represent the combination of the choice space. Then $A[c_{x1}][c_{x2}] \dots [c_{xN}]$ means the total time of the turns that player 1, 2, ..., N chose $c_{x1}, c_{x2}, \dots, c_{xN}$. In such situation, formula (10) will be simplified into

$$\prod_j P(X_{j,m} = h_{j,t} | X_i = c_k) = \frac{A[r_{1,t-1}][r_{2,t-1}] \dots [r_{i-1,t-1}][c_k][r_{i+1,t-1}] \dots [r_{N,t-1}]}{\sum_{k=1}^K A[r_{1,t-1}][r_{2,t-1}] \dots [r_{i-1,t-1}][c_k][r_{i+1,t-1}] \dots [r_{N,t-1}]}. \quad (15)$$

The time complexity of the prediction will reduce to $O(NK)$. The time complexity of the step of recording will be $O(1)$ (actually it is $O(N)$ in actual data structure), while the space complexity will rise to $O(K^N)$. The space complexity is acceptable when the K and N are relatively small.

The N -dimension array can just be satisfied in 1-step date-back, and its data structure will be complexity when the N becomes large. Here we can use hash table to make a more efficient record. A list of m -step data record is shown as follows:

$$\begin{aligned} & \{F_l, F_{l-1}, \dots, F_{l-m+1}\} \\ &= \{\{c_{x1}, c_{x2}, \dots, c_{xN}\}_{l-1}, \dots, \{c_{x1}, c_{x2}, \dots, c_{xN}\}_{l-m}\}. \end{aligned} \quad (16)$$

We can get its hash code through a hash function:

$$\text{hash}_l = \text{Ha}(\{F_l, F_{l-1}, \dots, F_{l-m+1}\}). \quad (17)$$

And the hash code provides an index of record array, which records the time that a specific situation happens. With the use of hash table, the complexity of prediction will still be $O(NK)$. The time complexity of the recording will be $O(1)$. And the space complexity will become $O(1)$ which is based on the device and not relevant to the N or K . Table 3 shows space and time complexity of different strategy.

4. Brief Step of the Algorithm

Firstly, we built an environment for the prisoner's dilemma game. Each player is asked to provide a strategy and update function. The program of the environment is as shown in Algorithm 2.

And the strategy and update function we provide is as shown in Algorithm 3.

5. Experimental Results and Analysis

5.1. The Performance of Bayes Model in the Double-Player Game. Four typical models were run against the Bayes model 10,000 times each; the total incomes of both players in each game were recorded. Figure 1 shows the overall incomes of both players recorded over 10,000 games comparing the proposed Bayes model with the other four typical models. Overall the Bayes model was more advantageous and achieved a higher score (overall income) than the other four. Of these other four typical strategy models, TFT performed best. It showed an equivalent overall income compared to that of the Bayes model and a higher income than all the others. For each game pair, this research presented two test results, each corresponding to one of the two stable score results from the selected game pair. Figure 2 reveals that the Bayes model earned a higher income than the random, Pavlov, and GTFT models. The income ratio to the GTFT model reached 6.6, while that with the TFT model also exceeded one. By examining the final income from repeated games, it was found that the Bayes model was more advantageous than the other four typical strategy models tested here.

In games repeated 10,000 times, the cases when the Bayes model scored 5, 3, 1, and 0 were statistically analyzed. As

TABLE 3: Time and space complexity of Bayes method with different data structures and some typical models.

	Space complexity	Time complexity of prediction	Time complexity of record
List	$O(LN)$	$O(KN^2L^2)$	$O(1)$
N -d array	$O(K^N)$	$O(NK)$	$O(N)$
Hash table	$O(1)$	$O(NK)$	$O(1)$
TFT	$O(N)$	$O(N)$	$O(N)$
GTFT	$O(N)$	$O(N)$	$O(N)$
Pavlov	$O(N)$	$O(N)$	$O(N)$
Random	$O(1)$	$O(1)$	$O(1)$

```

for  $t$  from 0 to MAX_TURN do
  for each player  $i$  do
    decision[ $i$ ]  $\leftarrow$  player[ $i$ ].strategy
  end
  feedback  $\leftarrow$  profit(decision)
  for each player  $i$  do
    player[ $i$ ].update(feedback)
  end
end

```

ALGORITHM 2

```

procedure strategy()
  if tem_turn < TURN_THRESHOLD then
    decision  $\leftarrow$  random(choice_space)
  else
    prediction_list  $\leftarrow$  predict(restore)
    for each choice  $c$  do
      profit[ $c$ ]  $\leftarrow$ 
        profit_expect( $c$ , prediction_list)
    end
  end
  decision  $\leftarrow$  arg(max(profit))
  return decision
end
procedure update( feedback ):
  storage.add(feedback)

```

ALGORITHM 3

shown in Figure 3, the scores of 5 and 1 represented a relatively large proportion. That is to say, the Bayes model was prone to defection. Analysis of Figures 1 and 3 implied that high scores mostly corresponded to cases scoring 5. Moreover, the results of each game competition showed that the incomes achieved by the Bayes model were higher when manifesting its tendency to defection.

Analysis of the overall income of both players in each game (Figure 4) showed that the overall income in the game with a TFT model was lower. According to the performance of the rivals in each game (Figure 3) and comparison with test result 2 (with the name end with "2"), it was noted that both strategy models in test result 1 (with the name end with "1") were less inclined to defection. Therefore, their overall

FIGURE 2: The income ratio of the Bayes model to other models in the games.

FIGURE 3: The distribution of Bayesian scores in each game.

FIGURE 4: The comparisons of overall game incomes.

income was higher. Since TFT is considered to be the model that can achieve more desired results amongst the four typical strategy models, games setting the TFT strategy model and the Bayes model in opposition were mainly investigated. In this game, the overall incomes of both sides were lower than those in other game competitions. This indicated that the game between the TFT strategy model and the Bayes model

both suffered losses. By studying the single game results from the Bayes model and the TFT model, it was found that the scores from the Bayes model were 1 (both sides simultaneously selected defection). This result suggested that, in any game between the Bayes model and the TFT model, defection appeared more frequently and represented a mark of the defection-prone tendencies of the two models.

5.2. The Performance of the Bayes Model in a Multiplayer Game. With regard to multiplayer games, this study jointly used four different strategy models to run against the Bayes model and the overall income from each model was recorded. With the methods described in Section 4, the income accruing to each player was distributed and the overall income was calculated. In this section, the decision method of TFT, Pavlov, and GTFT models differed slightly from those applied to the two-player game: in the event of the defection of one of the other players in the previous round, the rivals selected defection. In the following two-player game, the decision for the current round was made with the investigation of the decisions of players A and B in the previous round.

It can be deduced from Figure 5 that, in the multiplayer game, the Bayes model returned the highest overall income. In addition, the overall game situation implied that the proportion that the four typical strategy models selected cooperation was the highest. That is to say, in the game with four models, each treated cooperation as its main strategy (Figure 6).

5.3. Analysis of the Performance of the Bayes Model versus Normal Models. The normal model refers to the strategy models that are possibly encountered in real-life enactments of the game, simulated here using a natural model. The natural model was a model adopting a random strategy (that is, when faced with identical decisions from the previous round, the probabilities that the natural models selected cooperation were different). To verify that the strategy selected by the proposed Bayes model reaped more benefits in games versus the normal model, the game between them was repeated 500 times. In each game, there were 1,000 selections. It was an attempt to more comprehensively analyze the advantages and disadvantages of the Bayes model.

Figure 7 shows that the Bayes model performed better than most of natural models. Overall, the income from the Bayes model can reach approximately 3,000 and even 4,500 in individual extreme cases. The income ratio in Figure 8 was maximized at approximately 14, while most of the income ratios were above one.

In addition, the TFT model also conducted the games with the normal model: the result is shown in Figure 9. It was shown that the average income of the TFT model was approximately 2,500, which was lower than that of the Bayes model, while still equivalent to those of its other rivals.

5.4. The Performance of the Bayes Model When Run over Fewer Games. Since the Bayes model was a machine learning model, it needed a certain amount of data to guarantee its learning. Therefore, when there were fewer games, whether or

FIGURE 5: The overall income of each model after 10,000 times of multiplayer game.

FIGURE 6: Frequency occurrence of games in the one-shot game when the number of betrayals is varied.

not the Bayes model could achieve better game results should be considered.

In this study, the Bayes model was evaluated using fewer game times in competition with the TFT model 100 times with the game repeated 100 times. From Figures 10 and 11, we can see that the Bayes model was more advantageous.

5.5. Game Results from a PTFT Model Compared with the Other Models. The game models discussed above merely considered the attitude immediately after the selection of the previous step of both sides, while the PTFT model took account of the attitude during the selection of the previous three steps. Over 10,000 runs of the PTFT model against the other models, the incomes of each model are shown in Figure 12 which suggested that the Bayes model was disadvantageous over the game and gained neither more nor less than the TFT model (both models became trapped in the mutual defection deadlock). Moreover, the Pavlov model returned the lowest individual income, while the GTFT model gained the most.

This revealed one of the disadvantages of the Bayes model: if it failed to comprehensively consider all state characteristics that may appear in the game, it cannot obtain the optimum solution. In the current experiment, since the decision state steps of both sides considered by the Bayes model were set to one, the Bayes model was incapable of obtaining the optimum income result in the game against the PTFT model.

6. Conclusions

The authors regarded the prisoners' dilemma as an incomplete information game with unpublicized game strategies. In the research a machine learning model was constructed to solve problems in incomplete information games. Based on the Bayesian model, we can make the prediction of players'

FIGURE 7: The income of the Bayes model and normal models in the game.

FIGURE 8: The income ratio of the Bayes model versus the general model.

FIGURE 9: The income of the TFT model and normal models in the game.

choices to better complete the unknown information. And we suggested the hash table to make improvement in space and time complexity. We built a game system with several types of game strategy for testing. The experimental results show that the proposed Bayes model could obtain more desired game results compared with conventional typical strategy models in double- or multiplayer games 10,000 times. It was even believed that the Bayes model was slightly better than the acknowledged optimal strategy TFT model. In a game with more general single-step decision modeling and fewer games runs, the Bayes model also dominated. This result indicated that the naive Bayesian classification algorithm was feasible and effective at establishing the strategy model of an incomplete information game. It provided a novel idea for solving incomplete information game problems.

However, the results obtained by the naive Bayesian classification algorithm showed certain defects: it was unable to obtain the desired solution in the case of the decision ability of a rival beyond its estimation range. Therefore, it reduced

FIGURE 10: The incomes of the Bayes and TFT models after 100 games.

FIGURE 11: The average income of the Bayes and TFT models after 100 games.

FIGURE 12: The incomes of each model after 10,000 games.

FIGURE 13: The income of Bayes model with different depths of considered step in weight function when completing with random m -step strategy.

the applicability of the machine learning algorithm when encountering complex models. This should be the subject of future research.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (No. 61100148), and Project on the Integration of Industry, Education and Research of Guangdong Province (No. 2012B091100489).

References

- [1] O. Nomia, *Games with Incomplete Information*, Université Paris 1 Panthéon-Sorbonne, Paris, France, 1998.
- [2] J. C. Harsanyi, "Games with incomplete information played by "Bayesian" players. I. The basic model," *Management Science*, vol. 14, no. 3, pp. 159–182, 1967.
- [3] M. Zinkevich, M. Johanson, M. H. Bowling, and C. Piccione, "Regret minimization in games with incomplete information," *Advances in Neural Information Processing Systems*, vol. 2008, no. 20, pp. 1729–1736, 2008.
- [4] E. Alpaydin, *Introduction to Machine Learning*, The MIT Press, Cambridge, Mass, USA, 2004.
- [5] C. M. Bishop, *Pattern Recognition and Machine Learning*, Information Science and Statistics, Springer, New York, NY, USA, 2006.
- [6] X. G. Zhang, "Introduction to statistical learning theory and support vector machines," *Acta Automatica Sinica*, vol. 26, no. 1, pp. 32–42, 2000.
- [7] R. J. Hanson, B. Cheeseman, and P. Stutz, *Bayesian Classification Theory*, NASA Ames Research Center, Artificial Intelligence Research Branch, 1991.
- [8] M. Nowak and K. Sigmund, "A strategy of win-stay, lose-shift that outperforms tit-for-tat in the Prisoner's Dilemma game," *Nature*, vol. 364, no. 6432, pp. 56–58, 1993.
- [9] D. M. Kreps, P. Milgrom, J. Roberts, and R. Wilson, "Rational cooperation in the finitely repeated prisoners' dilemma," *Journal of Economic Theory*, vol. 27, no. 2, pp. 245–252, 1982.
- [10] D. W. K. Yeung, L. A. Petrosyan, and M. C. C. Lee, *Dynamic Cooperation: A Paradigm on the Cutting-Edge of Game Theory*, China Market Press, 2007.
- [11] R. Axelrod, "Effective choice in the prisoner's dilemma," *Journal of Conflict Resolution*, vol. 24, no. 1, pp. 3–25, 1980.
- [12] R. Axelrod and W. D. Hamilton, "The evolution of cooperation," *Science*, vol. 211, no. 4489, pp. 1390–1396, 1981.
- [13] J. H. Miller, "The coevolution of automata in the repeated prisoner's dilemma," *Journal of Economic Behavior and Organization*, vol. 29, no. 1, pp. 87–112, 1996.
- [14] W. H. Press and F. J. Dyson, "Iterated Prisoner's Dilemma contains strategies that dominate any evolutionary opponent," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 109, no. 26, pp. 10409–10413, 2012.
- [15] H. Lin and C.-X. Wu, "Evolution of strategies based on genetic algorithm in the iterated prisoner's dilemma on complex networks," *Acta Physica Sinica*, vol. 56, no. 8, pp. 4313–4318, 2007.
- [16] M. A. Nowak and K. Sigmund, "Evolutionary dynamics of biological games," *Science*, vol. 303, no. 5659, pp. 793–799, 2004.
- [17] J. W. Weibull, *Evolutionary Game Theory*, MIT Press, Cambridge, Mass, USA, 1997.
- [18] M. A. Nowak and R. M. May, "Evolutionary games and spatial chaos," *Nature*, vol. 359, no. 6398, pp. 826–829, 1992.
- [19] A. Cardillo, J. Gómez-Gardeñes, D. Vilone, and A. Sánchez, "Co-evolution of strategies and update rules in the prisoner's dilemma game on complex networks," *New Journal of Physics*, vol. 12, no. 10, Article ID 103034, 2010.
- [20] W.-B. Du, H.-R. Zheng, and M.-B. Hu, "Evolutionary prisoner's dilemma game on weighted scale-free networks," *Physica A: Statistical Mechanics and Its Applications*, vol. 387, no. 14, pp. 3796–3800, 2008.
- [21] H. Ohtsuki, C. Hauert, E. Lieberman, and M. A. Nowak, "A simple rule for the evolution of cooperation on graphs and social networks," *Nature*, vol. 441, no. 7092, pp. 502–505, 2006.
- [22] Y. Wang and C. Dang, "An evolutionary algorithm for global optimization based on level-set evolution and latin squares," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 5, pp. 579–595, 2007.
- [23] Y. Wang, Y.-C. Jiao, and H. Li, "An evolutionary algorithm for solving nonlinear bilevel programming based on a new constraint-handling scheme," *IEEE Transactions on Systems, Man and Cybernetics C: Applications and Reviews*, vol. 35, no. 2, pp. 221–232, 2005.
- [24] R. Selten and R. Stoecker, "End behavior in sequences of finite Prisoner's Dilemma supergames A learning theory approach," *Journal of Economic Behavior and Organization*, vol. 7, no. 1, pp. 47–70, 1986.
- [25] D. Y. Jiang, *Situation Analysis of Double Action Games with Entropy*, Science Press, New York, NY, USA, 2010.
- [26] P. A. Flach and N. Lachiche, "Naive Bayesian classification of structured data," *Machine Learning*, vol. 57, no. 3, pp. 233–269, 2004.