Using Kullback-Leibler Divergence to Model Opponents in Poker

Jiajia Zhang[1], Xuan Wang[2], Lin Yao[3], Jingpeng Li [1], Xuedong Shen [1]

[1]Intelligence Computing Research Center, Harbin Institute of Technology
Shenzhen Applied Technology Engineering Laboratory for Internet Multimedia Application
[2]Intelligence Computing Research Center, Harbin Institute of Technology
Public Service Platform of Mobile Internet Application Security Industry
[3]School of Electronics Engineering and Computer Science, Peking University
[1,2]C302, HIT Campus Shenzhen University Town, NanShan District, XiLi, Shenzhen 518055, P. R. China
[2]IER Building ,Soutn Area ,Shenzhen High-Tech Industrial Park .Shenzhen 518055, P. R.China
zhangjiajia_hit@163.com, wangxuan@insun.hit.edu.cn，1250047487@qq.com，lijingpengvip@vip.qq.com，1033013581@qq.com

Abstract

Opponent modeling is an essential approach for building competitive computer agents in imperfect information games. This paper presents a novel approach to develop opponent modeling techniques. The approach applies neural networks which are separately trained on different dataset to build K- model clustering opponent models. Kullback-Leibler (KL) divergence is used to exploit a safety mode on opponent modeling. Given a parameter d that controls the max divergence between a model's centre point and the units belong to it, the approach is proved to provide a lower bound of expected payoff which is above the minimax payoff for correctly clustered players. Even for the players that are incorrectly clustered, the lower bound can also be unlimited approximated with sufficient history data. In our experiments, agent with the novel model shows an improved classification efficiency of opponent modeling comparing with relative researches. And also, the new agent performs better when playing against poker agent HITSZ_CS_13 which participate Annual Computer Poker Competition of 2013.

Keywords: poker; imperfect information; opponent modeling; Kullback-Leibler divergence;

Introduction

Games can be classified as perfect or imperfect information conditions, which are based on whether or not players have the whole information of the game (Howard 1994). In imperfect information games, certain relevant details are withheld from the players. Poker is a typical interesting test-bed for artificial intelligence research in this area. It is a game of imperfect knowledge, where multiple competing agents must deal with risk management, agent modeling, unreliable information and deception, much like decision-making applications in the real world (Billings, Papp and Schaeffer 1998).

As a recognized approach of building a competitive poker player, the rationality of opponent modeling is based on that the "optimal strategies" in imperfect information conditions is actually randomized strategies (Kuhn 1950). And also, the opponents in the game are certainly not "optimal player", having idiosyncratic weaknesses like planning or execution errors (Archibald, Altman and Shoham 2010). These can be exploited to obtain higher payoffs than Nash value of the game (Southey, Bowling, and Larson eds. 2012). Approaches that discard opponent modeling usually follow a minimax strategy which supposes a worst case condition. The expected payoff is bounded by the minimax payoff. However, opponent modeling approaches provide higher expected payoffs than minimax payoff for its exploitation of opponents' weakness and specified strategies based on different characterized opponents. Thus, the problem of playing game safely, which means guaranteeing the lower bound of expected payoff regardless of the strategy used by the opponent, is a very important aspect in game system (Ganzfried, Sandholm 2012).

Represented by University of Alberta research group, many researchers study on opponent modeling approaches. In 1998, D. Billings explained how they implemented both specific and generic opponent modeling in Loki (Billings, Papp and Schaeffer 1998). This computer program was the first successful demonstration of opponent modeling improving the performance of a poker bot. In 2000, ANN (artificial neural network) method is applied on opponents modeling (Davidson, Billings, and Schaeffer. eds. 2000). In 2010, a method called group specific opponent model-

ing is proposed which created several different opponent models using K-model clustering (Van der Kleij 2010). ANN was applied to upper method and declared better performance (Fedczyszyn, Koszalka, and Pozniak-Koszalka 2012). The characteristic of k-model clustering opponent modeling is less dependency on the scale and accuracy of history opponent data besides some decrease of prediction accuracy. Thus, the novel approach proposed in this paper is motivated on the improvement of modeling accuracy and a way of expect payoff's lower bound calculation.

This paper is organized as following. Section 2 briefly introduces related works of opponent modeling in poker. Section 3 provides a modified opponent modeling approach recommended in this paper which is guided by KL divergence. Section 4 proved the safety of this approach. Given a parameter d that controls the max divergence of units in models, the algorithm is proved to provide a lower bound of expected payoff which is above the minimax payoff. To verify the performance of the new approach, section 5 shows the experiments results in practice. And finally, Section 6 gives our conclusions.

Related works

Neural network for predicting opponents' actions

As a popular and sophisticated approach, artificial neural networks (ANN) have been adopted to guide opponent modeling for many years. In order to predict an opponent's next action, history data are collected to train the network. By logging game contexts and the associated observed actions from poker games, training data was collected for a variety of different players. These players contain human players on internet game platform and computer players from history record of ACPC matches in past years.

The ANN's output nodes represent the network's prediction that an opponent will fold, call, or raise respectively. An output node can give a real value from 0 to 1. So by normalizing the output nodes, probability distribution or "Probability Triple" data structure are adopted. By training the network on all of history data, and using back-propagation to perform a gradient descent on the connection weights in the network, the network begins to successfully discover the importance of each input feature with regards to the opponent's decision process.

The structure of network for predicting opponent is illustrated in figure 1. In this network, 14 input nodes are used to describe current conditions and history data of the game. The input nodes can be classified as three types. First is about the actions frequency taken by players in this game. Second is about the current conditions of the game which contains the stage of game processes, the conditions of public cards and so on. The third group is about the history data statistics in our database.

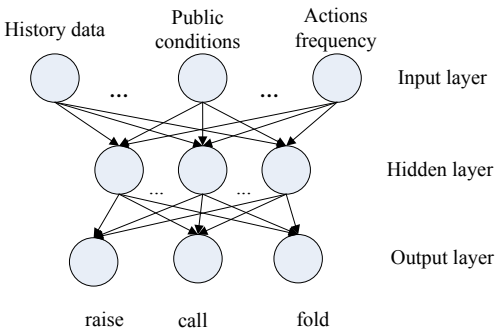

Figure 1. An Example ANN for opponent modeling

Kullback-Leibler divergence

In probability theory and information theory, Kullback–Leibler divergence is a non-symmetric measure of the difference between two probability distributions P and Q (Kullback and Leibler 1951). Specifically, the Kullback–Leibler divergence of Q from P, denoted DKL(P||Q), is a measure of the information lost when Q is used to approximate P. Formula 1 shows the definition of DKL.

$$D_{\mathrm{KL}}(P\|\,Q) = \sum_{i} \ln\left(\frac{P(i)}{Q(i)}\right) P(i) \tag{1}$$

In words, it is the expectation of the logarithmic difference between the probabilities P and Q, where the expectation is taken using the probabilities P.

For distributions P and Q of a continuous random variable, KL-divergence is defined to be the integral as formula 2 shows.

$$D_{\mathrm{KL}}(P\|\,Q) = \int_{-\infty}^{\infty} \ln\left(\frac{p(x)}{q(x)}\right) p(x)\mathrm{d}x \tag{2}$$

In the problem of Texas Hold'em poker, the prediction of opponents' strategy is based from the estimated probability of strategy triple S(r,c,f). Three terms separately donate the probability of opponents' "raise", "call" and "fold". Thus, suppose P and Q are two strategies in poker game, the KL divergence between them can be calculated as formula 3.

$$\begin{aligned} D_{\mathrm{KL}}(P\|\,Q) &= \ln\left(\frac{P(\mathrm{raise})}{Q(\mathrm{raise})}\right) P(\mathrm{raise}) \\ &+ \ln\left(\frac{P(call)}{Q(\mathrm{call})}\right) P(\mathrm{call}) \\ &+ \ln\left(\frac{P(\mathrm{fold})}{Q(\mathrm{fold})}\right) P(\mathrm{fold}) \end{aligned} \tag{3}$$

Researchers have revealed several properties about KL divergence (Seldin and Tishby 2010). One of them that support research of this paper can be illustrated as following.

Suppose p is the exact opponent's strategy, according to which the opponent's action are drawn. p' donates the predicted distribution based on opponent modeling result. |S| means the number of possible strategies that opponent may take, in poker it is 3. N donates the scale of empirical data. The KL divergence between p and p' in bounded by formula 4 with the probability at least m.

$$D_{\mathrm{KL}}(\mathrm{p}\|\mathrm{p}') \le \varepsilon(m) = \frac{(|\mathrm{S}|-1)\ln(\mathrm{N}+1) - \ln(1-m)}{N} \qquad (4)$$

Formula 4 reveals the relationship between KL divergences and scale of training data restricted by probability factor m. This guarantees the evaluable property of modeled units when the model process is guided by KL divergence.

Modified approach of opponent modeling

In this section, a novel algorithm for opponent modeling is presented, which is a modification of k-model clustering approach presented in (Van der Kleij 2010) and (Fedczyszyn, Koszalka, and Pozniak-Koszalka 2012). KL divergence is recommended and there are mainly two points of the advantages of novel approach. Firstly, KL divergence is used as a measurement that distributes units to certain clusters which improves the effectiveness of clustering process. Secondly, the novel approach presents a safe model that certificates the theoretical lower bound of expected payoff based on the prediction of clustered opponents.

Modified K-model clustering opponent modeling

The result mode of K-model approach can be realized by decision tree or neural network algorithm. Take neural network for example, the clustering system will build a series of neural networks matches each special group of opponents. Attention that the training data are pre-clustered for certain game states and the following cluster process are corresponded to one of them. In another word, if we treat the game for *n* different states and initial models number is *k*, a total of *n*k* networks, each for different clusters will be build in the whole process.

The clustering process can be processed as 4 steps in which *d* is fore assigned as the safety parameter.

Step 1: Random Assignment

In this step, each opponent in train data set is assigned to a random cluster.

Step 2: Training Neural Networks

In this step, k neural networks are trained based on the data of their matching cluster.

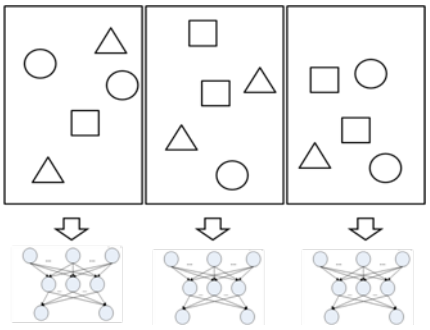

Figure 2. Random assignment and training steps

Step 3: Cluster rearrangement

This is the core step of the whole process. The opponents are re-assigned based on their KL divergence from the clusters.

Generally speaking, most effective clustering result is to form a similar distribution between clusters and units that clustered. For the purpose, player *q* is re-arranged based on following rules.

1. Choosing the cluster which has the minimum KL divergence from *q* as the candidate cluster.

The new assignment for a player is found by calculating the minimum KL divergence between the cluster's strategy distribution and the player's. That is, the new cluster for player *q* can be determined as follows:

$$Cluster\,(\mathrm{j}) = \underset{i\in\{0,\ldots k-1\}}{\operatorname{argmin}} D_{KL}(\mathrm{C}_j = \mathrm{i}\,\|\,\mathrm{P}_q\,|\,\mathrm{m}) \qquad (5)$$

Where C_j and Pq are the strategy distribution triples of cluster *j* and player *q* under condition *m*. k is the clusters' number built clusters in former step.

2. If the KL divergence between *q* and candidate cluster smaller than *d*, arrange *q* to candidate cluster. Otherwise, a new cluster is created and arranged *q* to it. Attention that the new cluster is only created once in one loop.
3.

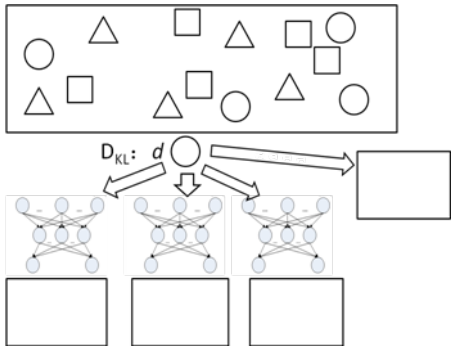

Figure 3. Rearrange player q to a new cluster

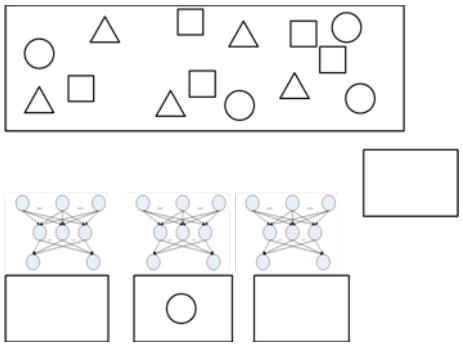

Figure 4. Rearrangement result of player q

Step 4: End condition judgment

In this step, the players' arrangements are compared with the initial conditions in step 2. If there is new cluster created in step 3 or exits player that is arranged differently from step 2's clusters, the system will go back to step 2 and all of the neural networks will be retrained with their new cluster. Otherwise, the clustering process is ended.

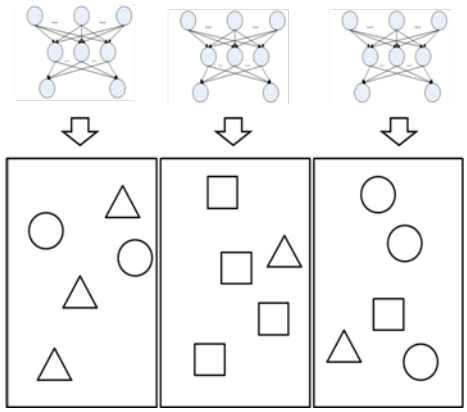

Figure 5. Example results after middle period iteration

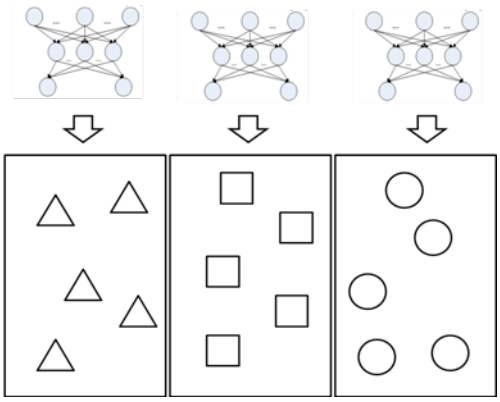

Figure 6. Example results in the end

Lower bound of expected payoffs

In the clustering process, safety parameter *d* is used as one of the criterion that decides to which cluster the opponents will be arranged. This section proves the theoretical meaning of *d* which provides the lower bound of expected payoffs when playing against opponents in certain cluster.

Table 1 shows a typical payoff matrix of a zero-sum game. Suppose the strategy set of player 1 and 2 is S {A, B}. The payoffs of the player 1 are listed in table based on different strategies of the two players.

Table 1. An example game's payoff matrix

game	player 2		
player 1		A	B
	A	V_{AA}	V_{AB}
	B	V_{BA}	V_{BB}

E_A and E_B donate the player 1's expected payoffs of A and B strategy. Approaches that discard opponent modeling usually need to make worst-case assumptions, e.g. following a minimax strategy. Such approaches are considered safe as their expected payoff is lower-bounded by the minimax payoff and E_A and E_B are calculated as following.

$$E_A = \min\{V_{AA}, V_{AB}\}$$
$$E_B = \min\{V_{BA}, V_{BB}\} \quad (6)$$

In opponent modeling system, they are calculated based on the prediction of opponent's behavior. Suppose player 2 are modeled by player 1 to have the strategy probability distribution as $P(p_A, p_B)$ in which p_A and p_B means the probability of player 2 adopts strategy A and B. E_A and E_B are calculated as following.

$$E_A = p_A V_{AA} + p_B V_{AB} \quad (7)$$
$$E_B = p_A V_{BA} + p_B V_{BB} \quad (8)$$

Generally speaking, opponent modeling system provides a higher expect payoff then minimax payoff. However, the safety of these expected payoffs cannot be evaluated with the uncertainty of the modeling precise.

However, the modified approach provided in this paper can be proved to keep a safety expect payoff which is higher than minimax payoff. Suppose player 2 is modeled as probability distribution P (p_A, p_B) and P' (p'_A, p'_B) is his exact distribution. Besides that the sum of p'_A and p'_B is 1, P' is also bounded safety parameter *d* and P (p_A, p_B).

Suppose c is the probability that player 2 is assigned in the correct cluster which is with distribution P. The KL divergence between player 2's real and modeled distributions is no more than *d* when he is correctly modeled, as formula 7 shows.

$$d \geq D_{KL}(p' \| p) = p'_A \ln \frac{p'_A}{p_A} + p'_B \ln \frac{p'_B}{p_B} \quad (7)$$

There are two pre-conditions for following derivation. First, the probability of the two strategies satisfies $p_A+p_B=1$. For the problems that have more than two strategies like poker, the strategies are integrated as two to satisfy

this condition. One donates the opponent's optimal strategies which led worst payoffs for us. The other is integrated by all left strategies. In another word, all strategies can be treated as two, which may lead the worst case or it may not.

Another precondition is the natural logarithm satisfies lnx<x-1for all x > 0 with equality if and only if x=1. Based these, the bound of P' can be deduced from formula 7 as following.

$$
\begin{aligned}
d \geq D_{KL}(p' \| p) &= p'_A \ln \frac{p'_A}{p_A} + p'_B \ln \frac{p'_B}{p_B} \\
&= p'_A \ln \frac{p'_A}{p_A} + (1 - p'_A) \ln \frac{(1 - p'_A)}{p_B} \\
&\geq p'_A \ln \frac{p'_A}{p_A} - (1 - p'_A) \frac{p'_A}{p_B} \\
&\geq p'_A (\frac{p'_A}{p_A} - 1) - (1 - p'_A) \frac{p'_A}{p_B} \\
&= p'_A (\frac{p'_A - p_A}{p_A} + \frac{p'_A - 1}{p_B}) \\
&= \frac{(p_A + p_A) p'^2_A - (p_A p_B + p_A) p'_A}{p_A p_B}
\end{aligned}
\tag{8}
$$

Solving the final result of formula 8, we can get the bound of probabilities in P'.

$$
\begin{aligned}
p'_A &\leq \frac{1}{2}(p_A p_B + p_A + \sqrt{(p_A p_B + p_A)^2 + 4 d p_A p_B}) = p'_{\max A} \\
p'_B &\leq \frac{1}{2}(p_A p_B + p_B + \sqrt{(p_A p_B + p_B)^2 + 4 d p_A p_B}) = p'_{\max B}
\end{aligned}
\tag{9}
$$

Thus we can get E_{BKL} as the lower bound of expected payoffs of player 2 when he is modeled as probability distribution P (p_A, p_B).

$$
\begin{aligned}
&E_A \geq E_{BKL} = \min\{p'_{\max A} V_{AA} + (1 - p'_{\max A}) V_{AB}, \\
&(1 - p'_{\max B}) V_{AA} + p'_{\max B} V_{AB}\} \\
&E_B \geq E_{BKL} = \min\{p'_{\max A} V_{BA} + (1 - p'_{\max A}) V_{BB}, \\
&(1 - p'_{\max B}) V_{BA} + p'_{\max B} V_{BB}\}
\end{aligned}
\tag{10}
$$

Thus far we can get the lower bound payoff when player 2 is correctly clustered. When he is assigned to a wrong cluster, the inequation 4 can be fit with a probability no less than *m*. In another word, the lower bound payoff E_{BKL} is also reliable with the probability no less than *m* which can be calculated based on formula 4:

$$
\begin{aligned}
d = \varepsilon(m) &= \frac{2\ln(N+1) - \ln(1-m)}{N} \\
\Rightarrow m &= 1 - \frac{N+1}{10^{Nd}}
\end{aligned}
\tag{11}
$$

Thus, the probability that E_{BKL} is the expect payoff lower bound can be considered as:

$$
\begin{aligned}
&\lim_{N \to \infty} P(\text{lowerbound} = E_{BKL}) \\
&= \lim_{N \to \infty} [c + (1 - c)\, m] \\
&= \lim_{N \to \infty} [c + (1 - c)(1 - \frac{N+1}{10^{Nd}})] \\
&= c + (1 - c) = 1
\end{aligned}
\tag{12}
$$

As formula 12 shows, when players are incorrectly clustered, the lower expected payoff bound is also reliable if there are sufficient history data. The only exception is when a player is absolutely new appeared and differs to all built clusters, the lower bound cannot be guaranteed any more.

Experiments

In our experiments, Texas Hold'em game is chosen as the experiments domain. This game is known as one of the most complex imperfect information games and good test bed for opponent modeling.

Clustering performance of new approach is tested based on a big data set of different player's strategies. The training and testing data are collected from human internet game platform and former matches of ACPC (Annual Computer Poker Competition). About 50 million rounds of game data are used and each of them contains strategies of different opponents under different game conditions.

All networks were trained using back propagation learning algorithm with learning rate set to 0.35. The training set is used 90% for training and the least 10% for testing. The initial number of clusters in K-model clustering process was set to divide players into 12 clusters (K=12). The safety parameter *d* is set as a changing variable to explore its influence on clustering efficiency. Attention that when d is set as a relative big value like 1, its influence is weakened and the modeling process will change to classical mode as research (Fedczyszyn, Koszalka, and Pozniak-Koszalka 2012) provides.

In related researches, there are three quality measures are popularly used to measure the performance of clustering approach (Van der Kleij 2010) (Fedczyszyn, Koszalka, and Pozniak-Koszalka 2012).

VPIP - Voluntary Put money Into Pot which tells us how often player plays a game preflop (does not fold preflop).

PFR - PreFlop Raise which informs how often player raises pre-flop.

AF - Aggression Factor which informs how aggressive player is.

The statistics results of experiment system with different set of *d* are shown in figure.

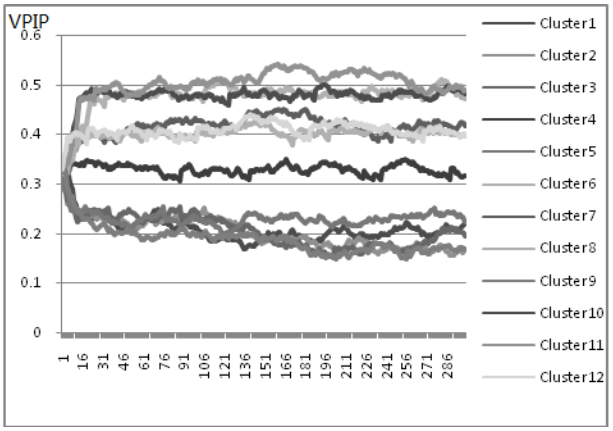

Figure 7 Average VPIP in clusters with d=1.

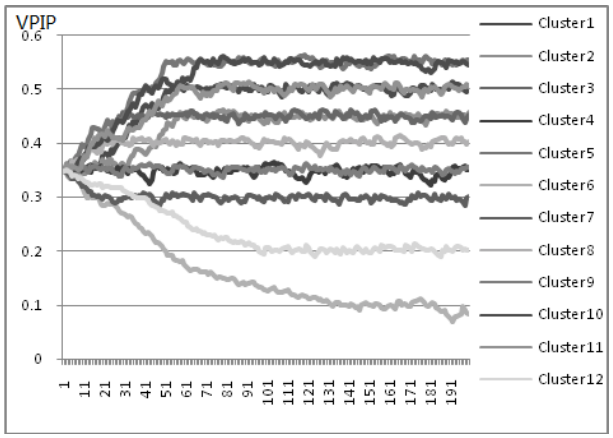

Figure 8 Average VPIP in clusters with d=0.15.

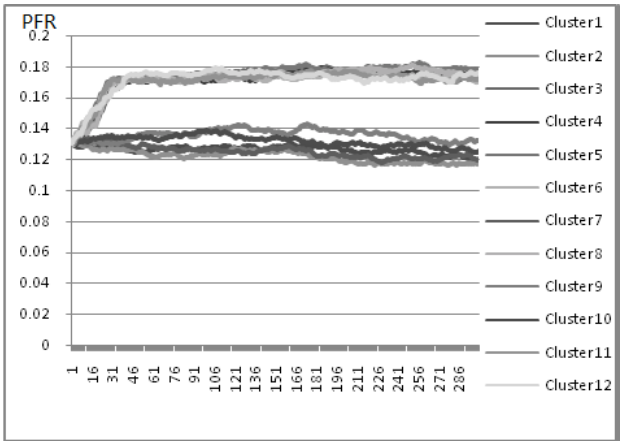

Figure 9 Average PFR in clusters with d=1.

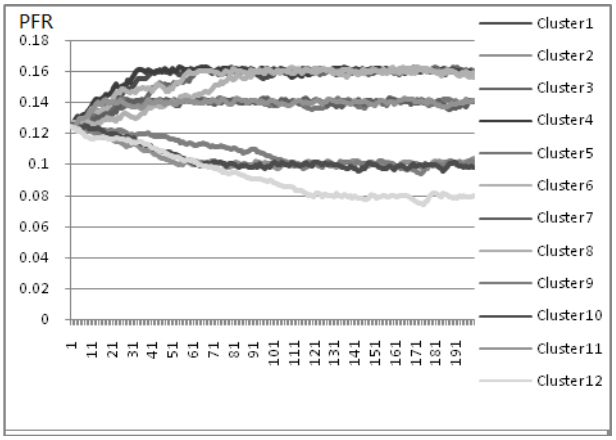

Figure 10 Average PFR in clusters with d=0.15.

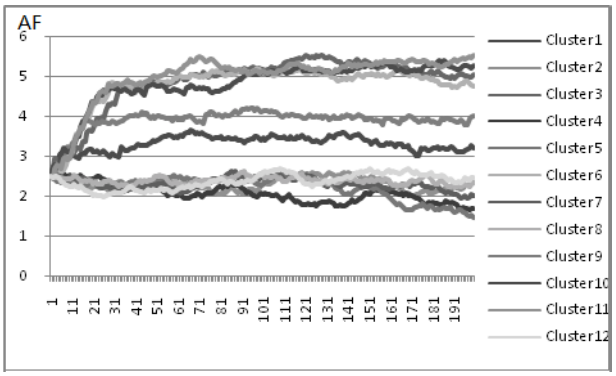

Figure 11 Average AF in clusters with d=1.

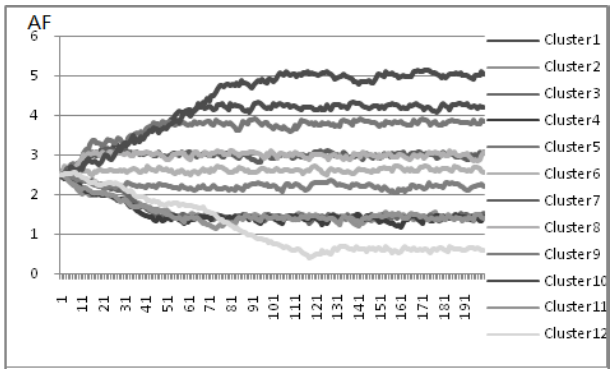

Figure 12 Average AF in clusters with d=0.15.

Figure 7 to figure 12 shows the modeling performance of the system with different *d*. the x axis shows the iteration times of clustering process and y axis shows the average value. The distinction degree among clusters is influenced by *d* based on a visual observation of VPIP, PFR and AF curves. When *d* is set too big, it will lose its restriction function in clustering process and the system will regress to normal k-model clustering process. However, when *d* is set too small, the training process will circulate for an un-

acceptable period for that there are exists units that cannot re-arranged in clustering step 3. When *d* is set to a proper value like 0.15, the opponents will be modeled into distinguished clustered as figure shows.

Another criterion of the effectiveness of opponent modeling is confusion matrix which shows the prediction precise of the modeled opponent's actions. Table 2 and table 3 show the confusion matrix of different d value where AC means the expected actions and TC means the true actions.

Each row in tables contains percentage value of correct or incorrect expectation of opponents' actions. The last column denoted as 'Total' contains percentage values of how many objects there were in a testing set while 'Total' row informs of how many objects were classified as objects of a given class. The cell, where 'Total' row and 'Total' column are crossed, contains percentage of correct classifications.

Table 2 Prediction accuracy of HITSZ_CS_13

AC\TC	Fold	Call	Raise	%
Fold	12.25	0.16	0.41	12.82
Call	3.50	47.65	10.28	61.43
Raise	2.37	7.15	16.22	25.74
%	18.12	54.96	26.91	76.12

Table 3 Prediction accuracy with d =0.15

AC\TC	Fold	Call	Raise	%
Fold	19.66	0.25	0.20	20.11
Call	1.96	52.31	11.39	65.66
Raise	0.05	2.18	12.00	14.23
%	21.67	54.74	23.59	83.97

Table 2 shows the prediction precise accuracy of our former system HITSZ_CS_13, which was participated in ACPC 13 and get the fourth rank in 3-player Limit Texas Hold'em . The total prediction correct rate without KL divergence's restriction is 76.12% which is also coordinate with relative research (Fedczyszyn, Koszalka, and Pozniak-Koszalka 2012). In another side, the recommend system guided by KL divergence shows an improved prediction performance which strictly contributes the game competitiveness of our poker system.

Conclusions

In this paper, a novel approach of k-model clustering opponent modeling guided by KL divergence is introduced. Based on the analysis and experiments, the modified approach shows an improved performance in opponent modeling process and further enhances the prediction precise of poker game system.

As a classic measure of the difference between two probability distributions, KL divergence effectively depicts the characters poker players. The effectiveness of k-model cluster opponent modeling is improved based on safety parameter *d* which is also provides the lower bound of expected payoffs. This conclusion can be established by the changing curve of different clusters' players in experiment's analysis.

However, there are also some problems in the new system. One of them is the application of KL divergence prolongs the period of k-model clustering approach. Especially when safety parameter *d* is set a too small, the clustering process cannot convergence in acceptable iteration times. This is the point that our further work will mainly focus on. And also, the neural network will be further studied. Classification characters that used for building more specified opponent models are also important for our research.

Acknowledgements

We would like to thank Xiao Ma for the theory direction for our system. We also appreciate Song Wu and Xinxin Wang for their fundamental work of the HITSZ_CS_13 system. And also, we gratefully thank the referees for their insightful researches.

Our works are supported by Shenzhen Applied Technology Engineering Laboratory for Internet Multimedia Application (Shenzhen Development and Reform Commission [2012]720), Public Service Platform of Mobile Internet Application Security Industry (Shenzhen Development and Reform Commission [2012]900).

References

Bampton, H. J. 1994b. Solving imperfect information games using the Monte Carlo heuristic, Master diss.. University of Tennessee, USA.

Billings, D.; Papp, D.; and Schaeffer, J. eds. 1998. Opponent modeling in poker. In *Association for the Advancement of Artificial Intelligence* : 493-499. AAAI Press

Kuhn, H. W. 1950a. A simplified two-person poker. In *Annals of Mathematics Studies* 24: 97–103.

Archibald, C.; Altman, A.; Shoham, Y. 2010. Success, strategy and skill: an experimental study. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems* 1(1):1089-1096. International Foundation for Autonomous Agents and Multiagent Systems.

Southey, F., Bowling, M. P., Larson, B., Piccione, C., Burch, N., Billings, D., and Rayner, C. 2012a. Bayes' bluff: Opponent modelling in poker. *arXiv preprint arXiv:1207.1411*.

Ganzfried, S.; Sandholm, T. 2012. Safe opponent exploitation. In *Proceedings of the 13th ACM Conference on Electronic Commerce* : 587-604. ACM Press.

Davidson, A.; Billings, D.; and Schaeffer, J. eds. 2000. Improved opponent modeling in poker. In *International Conference on Artificial Intelligence, ICAI'00*: 1467-1473. ICAI Press.

Van der Kleij, A. A. J. 2010b. Monte Carlo Tree Search and Opponent Modeling through Player Clustering in no-limit Texas Hold'em Poker. Master diss., University of Groningen, The Netherlands.

Fedczyszyn, G., Koszalka, L., and Pozniak-Koszalka, I. 2012. Opponent Modeling in Texas Hold'em Poker. In *Computational Collective Intelligence. Technologies and Applications*: pp. 182-191. Springer Berlin Heidelberg Press.

Kullback, S., Leibler, R. A. 1951a. On information and sufficiency. *The Annals of Mathematical Statistics*: 79-86.

Seldin, Y., and Tishby, N. 2010a. PAC-Bayesian Analysis of Co-clustering and Beyond. *Journal of Machine Learning Research* 11:3595 – 3646.