Building a No Limit Texas Hold'em Poker Agent Based on Game Logs using Supervised Learning

Luís Filipe Teófilo and Luís Paulo Reis

Departamento de Engenharia Informática, Faculdade de Engenharia da Universidade do Porto, Portugal
Laboratório de Inteligência Artificial e Ciência de Computadores, Universidade do Porto, Portugal
{luis.teofilo,lpreis}@fe.up.pt

Abstract. The development of competitive artificial Poker players is a challenge to Artificial Intelligence (AI) because the agent must deal with unreliable information and deception which make it essential to model the opponents to achieve good results. In this paper we propose the creation of an artificial Poker player through the analysis of past games between human players, with money involved. To accomplish this goal, we defined a classification problem that associates a given game state with the action that was performed by the player. To validate and test the defined player model, an agent that follows the learned tactic was created. The agent approximately follows the tactics from the human players, thus validating this model. However, this approach alone is insufficient to create a competitive agent, as generated strategies are static, meaning that they can't adapt to different situations. To solve this problem, we created an agent that uses a strategy that combines several tactics from different players. By using the combined strategy, the agent greatly improved its performance against adversaries capable of modeling opponents.

Keywords: Poker; Machine Learning; Supervised Learning; Opponent Modeling; Artificial Intelligence.

1 Introduction

Poker is a game that is increasingly becoming a field of interest for the AI research community, on the last decade. Developing an agent that plays Poker presents a completely different challenge than developing agents for other games like chess or checkers. In these games, the players are always aware of the full state of the game. This means that these problems can be solved by using deep Minimax decision trees and game state classification heuristics. Notable results were achieved in this type of games such as the Deep Blue, which was the first to computer to beat a Chess champion in a series of games [1].

Unlike Chess, the Poker's game state is partially hidden because each player cannot see the opponents' cards. Therefore, it's much more difficult to create and analyze a decision tree. Poker is also stochastic game i.e. it admits the element of chance, making the outcome of the game unpredictable.

The announced characteristics of Poker make it essential to define mechanisms to model the opponents, before making a decision. By identifying the opponents playing style, it is possible to predict their possible actions and therefore make a decision that has better probability of success [2, 3].

The main goal of this work is to determine on how strategies used in the past by good human players can be used to support the creation of an artificial Poker player. This work is divided in the following steps:

- Extract a significant amount of games between human players;
- Extract information about the players;
- Define the classification problem: game state variables that can influence player's decision and the possible outcomes;
- Create a strategy by classifying game state instances, using supervised learning algorithms;
- Create and test an agent that uses the classifier and follows the strategy.

The rest of the paper is organized as follows. Section 2 briefly describes the game of poker. Section 3 describes related work about opponent modeling in Poker and some approaches followed earlier to create Poker agents. Section 4 describes the characteristics of the dataset that was used in this work. Section 5 presents the classification problem and the comparison of the supervised learning algorithms that were used to solve it. Section 6 presents some details about the implementation of the agent that uses the human strategy. Section 7 describes the results obtained by the agent. Finally, section 8 presents the paper main conclusions and some pointers for future work.

2 Texas Hold'em Poker

Poker is a gambling game where the players bet that their hand is stronger than the hands of their opponents. Every bet made by a player goes into the pot and at the end the winning player collects the pot. The winner is the last standing player or the player that has the best hand. Poker is a generic name for literally hundreds of games, with similar rules [4], called variants. This work is focused on the variant Texas Hold'em Poker, which is nowadays the most popular one.

2.1 Hand Ranking

A poker hand is a set of five cards that identifies the score of a player in a game of poker. The hand rank at any stage of the game is the score given by the 5 card combination composed by player cards and community cards that has the best possible score. This means that if more than 5 cards are available, some will be ignored.

The possible hand ranks are (from stronger to weaker ranks): Royal Flush (top sequence of same suit), Straight Flush (sequence of same suit), Four of a Kind (4 cards with same rank), Full House (Three of a Kind + Pair), Straight (sequence of cards), Three of a Kind (3 cards with same rank), Two Pair, One Pair (2 cards with same rank) and Highest Card (when does not qualify to any other ranks).

2.2 No Limit Texas Hold’em

No Limit Texas Hold’em is a Poker variation that uses community cards. At the beginning of every game, two cards are dealt for each player – pocket cards. A dealer player is assigned and marked with a dealer button. The dealer position rotates clockwise from game to game. After that, the player on the left of the dealer position posts the small blind (half of the minimum bet) and the player on the left of that one posts the big blind (minimum bet). The first player to talk is the one on the left of the big blind position.

After configuring the table, the game begins. The game is composed by four rounds (Pre-Flop, Flop, Turn, River) of betting. In each round the player can execute one of the following actions: Bet, Call, Raise, Check, Fold or All-In.

At any game round, the last standing player wins the game and therefore the pot. If the River round finishes, the player that wins is the one with the highest ranked hand.

This variation of Poker is also No Limit, which means that the players are allowed to bet any amount of chips, above the Bet or Raise value.

3 Related Work

Most noticeable achievements in Computer Poker are from the Poker Research Group [5] from University of Alberta mostly on the variant Limit Texas Hold’em. One of the most significant publications of the group is Darse Billings PhD thesis [6] where he presents a complete analysis of the evolution of artificial poker agent architectures, demonstrating strengths and weaknesses of each architecture both in theory and in practice.

Other significant publication is [7] where a perfect strategy was defined for a very simple variant of Poker (Rhode Island). Another near-perfect strategy was achieved in [8] for Heads-Up No Limit variant, using the Nash Equilibrium Theory. More recently, it should be emphasized the article [9], which describes a new and more effective technique for building counter strategies based on Restricted Nash Response. Some achievements were also made using machine learning classifiers like in [10] where were studied evolutionary methods to classify in game actions.

Despite all the breakthroughs achieved by known research groups and individuals, no artificial poker playing agent is presently known capable of beating the best human players.

3.1 Opponent Modeling in Poker

A large number of opponent modeling techniques are based on real professional poker players' strategies, such as David Sklansky, who published one of the most renowned books on poker strategy [4]. In his book Sklansky says that Poker players can be classified in terms of tightness and aggressiveness. A player is tight if it plays 28% or less hands, and loose if it plays more than 28% of the times. With regard to aggressiveness, the player is aggressive if it has an aggression factor (1) over 1.0; otherwise he is a passive player.

$$Agression\ Factor = \frac{Num\ Raises}{Num\ Calls} \tag{1}$$

With these two parameters, Sklansky defined groups of players (Sklansky' groups) that play similar hands in a similar way.

3.2 Poker Hand Rank Evaluation

Hand rank evaluation consists in checking if a given Poker hand is better than another. Usually, the Hand Rank Evaluator takes a poker hand and maps it to a unique integer rank such that any hand of equal rank is a tie, and any hand of higher rank wins. For a poker AI, it is absolutely critical to have the fastest hand evaluator possible [11]. Any poker agent may have to evaluate thousands of hands for each action it will take.

The fastest known evaluator is TwoPlusTwo Evaluator [12], which can evaluate about 15 millions of hands per second.

3.3 Odds Calculation

Evaluating the rank of the hand, which was discussed above, is about giving a score to a set of cards. A Poker AI in reality does not directly use the hand evaluators, because it does not know the opponent's cards, so there is no group of hands to compare. A Poker AI performs odds calculation, which consists on the prediction of its own hand success. For that purpose, it evaluates its own hand score and compares with possible opponent's hands scores. This prediction is used to help measuring the risk of an action.

There are various ways to determine the odds:

- Chen Formula [13]: this formula can determine the relative value of a 2 card hand;
- Hand Strength [6, 11]: determines how many hands are better than our, taking into account the number of opponents;
- Hand Potential [6, 11]: The hand potential is an algorithm that calculates PPOT and NPOT. The PPOT is the chance that a hand that is not currently the best improves to win at the showdown. The NPOT is the chance that a currently leading hand ends up losing. Therefore, they are used to estimate the flow of the game;
- Effective Hand Strength [6, 11]: Combines the hand strength and the hand potential formulas.

3.4 Agent development tools

There are some software tools that can aid the development of Poker agents. Most notable is the Meerkat API [11] which easily allows the creation of Poker agents. The Meerkat agents can be tested using game simulators that support this API, like Open Meerkat Test Bed [14].

4 Poker Data Extraction and Analysis

To create models of human poker players, the first step is to extract a great amount of Poker games to analyze the actions of the players. The chosen data source is composed by game logs from money games in online casinos.

The Poker game logs represent the history of actions made by the players during the games on a given Poker table. These files don't represent the state of the game. Therefore to analyze the game it is necessary to replay the game with the same movements and card distribution.

Obtaining information from these documents is difficult since these files don't typically have an organized structure, making it difficult to parse the information. Moreover, there is no standard to represent game movements: each casino client has its own representation of logs. For this reason, to combine data from multiple sources, a new parser is needed for each game log format.

The package of game logs that was used in this work can be found here [15]. Some characteristics of the game logs are on table 2.

Table 1. Game logs characteristics

Characteristic	Value
Number of games	51.377.820
Number of players	158.035
Number of showdowns	2.323.538
Number of players with 500 or more showdowns	183

To characterize an action it is essential to know the cards that the player had. As it can be seen on table 1, the percentage of games in which card exposal occurs is very small, and only 183 players showed their cards more than 500 times. These were the players that were selected for this work.

After obtaining the game data, we selected the best players available in order to learn good tactics. For this task, a player list was generated containing some information about each player. The criteria used to choose a good player was its earnings as, for instance, a player with high negative earnings probably don't has a good tactic.

Table 2. Characteristics of the extracted players

Name	Game Count	Number of Shows	Earnings
John	15.763	638	1.003$
Kevin	20.660	838	30$
David	77.598	2.103	14.142$
Jeff	33.257	882	-4.945$

It should be noted that a player with negative earnings (Jeff) was included for testing purposes, to check if a tactic generated from a theoretical bad player loses against a tactic from a theoretical good player.

5 Learning Game Strategies

To learn the players‘ tactics supervised learning algorithms were used. In order to facilitate both the learning process and the implementation of the agent, we used WEKA[17]. The characteristics used to classify the players’ tactics were based on game state variables that might influence the players’ decisions during the game: position in table, money, last opponent’s action and classification of last opponent. The following code represents the ARFF file structure associated with each player.

```
@relation poker
@attribute positionScore numeric
@attribute effectiveHandStrength numeric
@attribute percentageMoneyToBet numeric
@attribute percentageMoneyOnTable numeric
@attribute possibleWins numeric
@attribute didLastPlayerRaise numeric
@attribute didLastPlayerCall numeric
@attribute isLastPlayerAggressive numeric
@attribute isLastPlayerTight numeric
@attribute action {call, raise5%, raise10%, ...}
```

Only numerical attributes were used in this model. The boolean attributes were converted to 0 or 1 whenever the value was true or false. The last attribute is nominal and it represents the action that was taken by the player, being for that the class attribute. The defined model returns an action, based on all the other attributes. Another important fact to note is that each player has four ARFF files associated with it, each one representing a game round (Pre-Flop, Flop, Turn, River). This is because tactics used during the game tend to be different in each game round, due to factors such as the varying number of community cards available.

Different classifiers were tested to build this model. The classifiers that obtained a smaller average error, using tenfold cross validation, were search trees, more particularly Random Forest Trees (Fig 1). The errors were relative high as was expected as players tend to change tactic during the game, making it difficult to find patterns on the data.

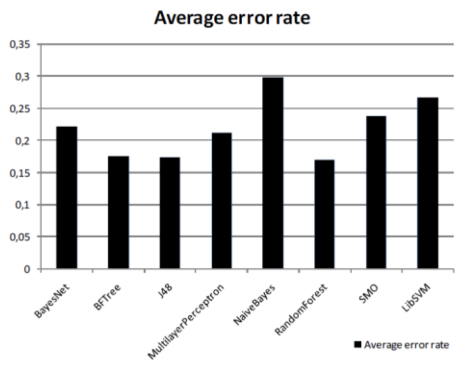

Fig. 1. Average classifier error rate.

The classifiers error rate was also analyzed per round (Fig. 2). It was found that the error is much higher in Flop and Turn rounds than in Pre Flop and River rounds. This was expected for Pre-Flop round, since there are no communities cards, the tactics tend to be simpler. As for River round, this can be explained by the lower number of players that reach this round.

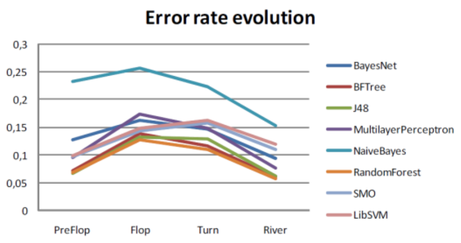

Fig. 2. Average classifier error rate evolution.

6 Agent Implementation

The agent was implemented using the Meerkat API [11]. The way the agent chooses the action is based on the current game state and can be divided in the following steps.

- First, the agent determines which actions can be taken;
- The agent changes the tactic based on the game state;
- It uses the current tactic classifier to predict the action;
- If the prediction is null i.e. the classifier can't find any action that suits the situation, the agent checks if possible, otherwise it folds.

Regarding the action prediction, the classifiers weren't taught how to fold, because only hands with showdown were used. To solve this, we defined a criterion to fold some hands. The defined criterion was that if the Effective Hand Strength [11] is below 50%, then the agent has a probability of folding the hand equal to its tightness level. The agent will choose the class that has better acceptance level on the current game state. If that class acceptance level is below a defined minimum, the agent folds.

7 Tests and Results

Four types of tests were used to validate this approach: behavior tests, tests between generated tactics, tests against other agents previously developed and strategy tests. The tests were made on Open Meerkat Test Bed, with 1.000 games and table seat permutation.

The results of the behavior tests are on table 3. The agent approximately conserved the real human Sklansky's classification thus conserving its behavior.

Table 3. Behaviour tests

Name	Agent AF	Real player AF	Agent tightness	Real player tightness	Agent classification	Real player classification
John	4,77	5,12	0,32	0,31	Loose Aggressive	
Kevin	1,55	1,49	0,25	0,23	Tight Aggressive	
David	16,04	13,22	0,19	0,19	Tight Aggressive	
Jeff	9,42	8,40	0,44	0,33	Loose Aggressive	

Fig. 3 presents the bankroll evolution in a series of games between the agents. The humans that won more money in real life generated an agent with better results.

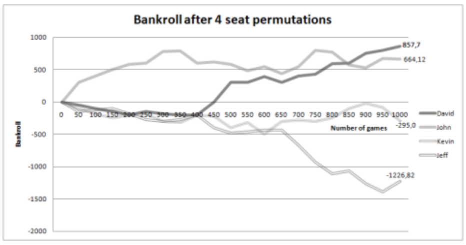

Fig. 3. Game between the players on table

Next, we tested John agent against HSB Bot (an agent that chooses his actions based on its pocket hand strength) (Fig. 4). John clearly won the match by a large margin. John was also tested against an always call agent, achieving similar results.

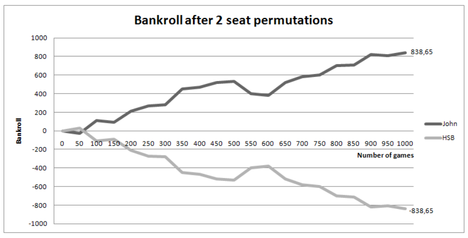

Fig. 4. John VS HSB

Finally, we tested John against MCTS Bot [18]. The agent was totally defeated (Fig. 5), because the MCTS is capable of opponent modeling.

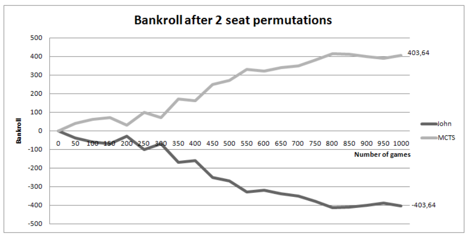

Fig. 5. John VS MCTS Bot

The results improved a lot by using a strategy that combines multiple tactics of generated agents (Fig. 6). The strategy changes the tactic when losing money, confusing the opponent modeling mechanisms. We can observe some cycles on the chart due to this fact. When the agent starts losing money, it changes its tactic, and after a little while it starts winning again, until the moment that the MCTS discovers the new tactic.

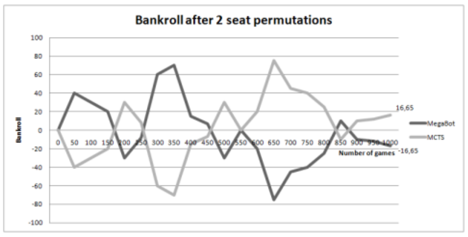

Fig. 6. Combined strategy VS MCTS

8 Conclusions and Future Work

There is still a long way to go to create an agent that plays poker at the level of the best human players. This research presented an approach based on supervised learning methodologies, where the agent relies on pre learned tactics based on past human experience to decide upon its actions. The created agent is not competitive against good poker players that are capable of opponent modeling. However the results greatly improved after combining various tactics, which means that an agent should not use a static tactic in a game like poker.

This approach can be promising for good human players, since they can create an agent based on their logs that will play like them, autonomously.

The generated agents could be improved in the future, by defining a more complex player model or specifying better heuristics to change tactic along the game. In the future, the system should also be tested against human players.

This approach might also be used in other incomplete information games by defining the game state variables of that particular game.

Acknowledgments. I would like to thank to Fundação para a Ciência e a Tecnologia for supporting this work by providing my PhD Scholarship SFRH/BD/71598/2010.

References

1. Hsu, F-H.: Behind Deep Blue: Building the Computer that Defeated the World Chess Champion. Princeton University Press (2002)
2. Billings, D., Papp, D., Schaeffer, J., Szafron, D.: Opponent modeling in poker. In: AAAI '98/IAAI '98, Madison, Wisconsin, United States. American Association for Artificial Intelligence, pp 493-499 (1998)
3. Davidson, A.: Opponent modeling in poker. M.Sc., University of Alberta, Edmonton, Alberta, Canada (2002)
4. Sklansky, D.: The Theory of Poker: A Professional Poker Player Teaches You How to Think Like One. Two Plus Two (2002)
5. Computer Poker Research Group Homepage, http://webdocs.cs.ualberta.ca/~games/poker/
6. Billigs, D.: Algorithms and Assessment in Computer Poker. Ph.D., University of Alberta, Edmonton, Alberta (2006)
7. Gilpin, A., Sandholm, T.: Better automated abstraction techniques for imperfect information games, with application to Texas Hold'em poker. In: 6th international joint conference on Autonomous agents and multiagent systems, Honolulu, Hawaii. ACM, pp 1-8 (2007)
8. Miltersen, P.B., Sørensen, T.B.: A near-optimal strategy for a heads-up no-limit Texas Hold'em poker tournament. In: 6th international joint conference on Autonomous agents and multiagent systems, Honolulu, Hawaii. ACM, pp 1-8 (2007)
9. Johanson, M., Bowling, M.: Data Biased Robust Counter Strategies. In: Twelfth International Conference on Artificial Intelligence and Statistics, April 16-18, 2009, Clearwater Beach, Florida, USA, pp 264-271 (2009)
10. Beattie, B., Nicolai, G., Gerhard, D., Hilderman, R.J.: Pattern Classification in No-Limit Poker: A Head-Start Evolutionary Approach. In: 20th conference of the Canadian Society for Computational Studies of Intelligence on Advances in Artificial Intelligence, Montreal, Quebec, Canada. Springer-Verlag, pp 204-215 (2007)
11. Felix, D., Reis, L.P.: An Experimental Approach to Online Opponent Modeling in Texas Hold'em Poker. In: 19th Brazilian Symposium on Artificial Intelligence: Advances in Artificial Intelligence, Savador, Brazil. Springer-Verlag, pp 83-92 (2008)
12. Poker Bot Artificial Intelligence Resources. http://spaz.ca/poker/
13. Chen, B., Ankenman, J.: The Mathematics of Poker. Conjelco (2006)
14. Félix, D., Reis, L.P.: Opponent Modelling in Texas Hold'em Poker as the Key for Success. In: ECAI 2008: 18th European Conference on Artificial Intelligence, Amsterdam, The Netherlands, The Netherlands. IOS Press, pp 893-894 (2008)
15. Meerkat Opentestbed, http://code.google.com/p/opentestbed/
16. Out Floppe Poker Q & A: Obfuscated datamined hand histories, http://www.outflopped.com/questions/286/obfuscated-datamined-hand-histories.
17. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H. The WEKA data mining software: an update. SIGKDD Explor Newsl 11 (1):10-18 (2009)
18. Broeck, G., Driessen, K., Ramon, J.: Monte-Carlo Tree Search in Poker Using Expected Reward Distributions. In: 1st Asian Conference on Machine Learning: Advances in Machine Learning, Nanjing, China. pp 367-381 (2009)