

A SURVEY ON ARTIFICIAL INTELLIGENCE IN STOCHASTIC GAMES OF IMPERFECT INFORMATION: POKER.

Deniz Dizman

Abstract – Poker is a stochastic game of incomplete information which contains elements that hold a challenge for artificial intelligence agents due to the nature of the game. These challenges are examined along with methods used to solve these problems and different approaches for building a strong poker AI agent.

Introduction

The domain of poker with its well defined rules creates an opportunity to investigate some scientifically challenging issues such as deliberately given misleading information, making decisions based on imperfect (partial) information, modeling opponents to figure out counter strategies. Poker also stands aside from classical computer games like chess or checkers due to the fact that poker is a game of multiple players, with stochasticity (random events distributed over a known probability), imperfect information (bits of information are hidden from players), partial observability (some information may never be revealed to the players). Taking into account the factor of deception, opponent modeling and uncertainty traditional methods of artificial intelligence in games cannot cope with the demands of poker. This paper will conduct a literature survey of the current methods used to create artificially intelligent agents that will respond to this problem.

The first part will define the problem domain and introduce the game of poker and variants. The second part will examine heuristic and simulation based approaches for creating a poker playing agent. The third part will discuss game theoretic methods. The fourth part focuses on opponent modeling for action prediction. The last part will explain agent evaluation and conclude the paper.

Stochastic games

In game theory stochastic games are dynamic games with transition probabilities played by one or more players [15]. The game progresses in stages. At the beginning the game is in a defined state and players choose from a set of available actions and the game continues into another state which depends on the previous state and the actions chosen by the players. The game continues state transitions for finite or infinite number of times. Each player receives a payoff according to their actions. The total payoff is usually the cumulative payoff from all stages of the game. Stochastic games are formed of the tuple $(n, S, A_{1..n}, T, R_{1..n})$ where n is the number of agents, A_i is the action state available to agent i , T is the transition function $S \times A \times S \rightarrow [0,1]$ and R_i is the reward function for agent i with $S \times A \rightarrow R$ [17]. For each $s \in S$ let $A(s)$ denote the vector $(A_1(s), A_2(s), \dots, A_n(s))$. For each $s, t \in S$ and $a \in A(s)$, $p_{s,t}(a)$ denotes the probability of the transition from state s to state t when all the players play their component action. At each state s the player i can choose any probability distribution σ over the actions in $A_i(s)$. Let $\sigma(a_i)$ denote the probability that action a_i is selected. Let $a = (a_1, \dots, a_n)$ and define $\sigma(a) = \prod_i \sigma(a_i)$. Then the definition of the transition function extends to $p_{s,t}(\sigma) = \sum_{a \in A(s)} [\sigma(a) p_{s,t}(a)]$ [18]. A policy π for player i is to choose probability distributions over actions in $A_i(s)$ at state s . If we denote the current state at time t then the goal of agent i is to maximize $\sum_{t=0}^{\infty} \gamma^t r_i(s_t)$ [18] where $0 < \gamma < 1$ is the discount rate. Stochastic games are an application of Markov Decision Processes to multiple agents. Stochastic games can be in the

form of zero-sum games and non zero-sum game according to the results of the sum of the rewards. If this sum is equal to zero the game is said to be a zero-sum stochastic game. Poker is a zero sum stochastic game due to the fact that the winners reward is the money from all of the losing players. Tournaments are extremely popular in poker where each entrant pays an amount of X for a chip stack of Y. If a player loses all of his chips he is eliminated from the tournament. The blinds increase based on time or hand number and the first 7 players to be eliminated lose their entrance fee, while the first player receives %50 of all the entrance fees, the second %30 and the third %20. Since the chips have no monetary value, the tournament can be regarded as a stochastic game with states that correspond to a vector of the players chip stack sizes.

Poker and its forms

Poker is a class of games. There are around a 100 variations of poker, which have different rules among themselves. Poker is played with two or more players from a standard deck of cards with or without the joker and involves a series of games in which players compete to beat all of the opponents by forming the highest valued five card hand. Each possible hand is associated with a category that is ranked by the possibility of the hand to be made, where hands with the lower possibility are ranked as stronger. Players wager against each other during a predetermined amount of betting rounds and if the game leads to a showdown where the players still in the game reveal their hands, the player with highest ranking hand wins the hand and takes the money wagered in the pot. Not all games may lead to a showdown and the players revealing their cards. During the bets players may choose not to match the bet made by a player in which case they are said to "fold" their cards. If all players but one should fold the remaining player will take the pot without being challenged.

The variant of poker called Texas Hold'em is of interest to us because of the strategical elements involved in the game and the factor of luck is the least effective. The game of Texas Hold'em poker can be played with two to ten players.

Each player is dealt two face down cards called the hole cards and during the progression of the game community cards are dealt face up on the board which the players use to form the best poker hand they can combining the community cards with their pocket cards.

Each game stage (explained below) consists of a combination of the following actions:

Actions

Fold: The player doesn't match the bet made by a previous player and forfeits any chances of winning the pot

Call: The player matches the highest bet made by a previous player and wagers the money into the pot. If no bet was made by a previous player this action is called a "check"

Bet: The player wagers a new bet into the pot which the other players must match to continue playing. If the player bets even though a previous player had already bet this action is called a "raise" and can be thought as if the player had called the previous bet and raised to a new amount.

Game stages

Preflop: Before each game a dealer is selected and the two seats to the left of the dealer are the small and big blinds and bet half and a full minimum bet respectively without receiving any cards. After the dealer has dealt the hole cards a round of betting begins.

Flop: The dealer deals three public cards face up to the board which the players will use to form their poker hand. The player to the left of the dealer begins a new betting round.

Turn: The dealer deals an additional public card and the betting round begins with the player to the dealers left talking first.

River: The dealer deals the final public card and the betting rounds begins with the player to the dealers left.

Showdown: All the players still in the game reveal their hands and the player with the highest ranking hands wins the game and takes the pot. In the case of a tie, the pot split evenly among the winning players.

Variants of Texas Hold'em

There are different versions of Texas Hold'em poker differing in the wagering allowed during the game.

In the limit version of the game players are allowed to make a fixed bet during each betting round. A \$10/20 limit game denotes that the value of the bet at the pre-flop and flop round is \$10 and \$20 in the river and turn round. Small blinds and big blinds are half a bet and a full bet respectively. In no-limit Texas Hold'em the blinds are determined before starting the game and the players may bet any amount they wish up their entire bankroll. During this survey we will concentrate on two player limit Texas Hold'em.

Heuristic and Simulation based methods

Heuristics based approaches provide a simple but crude approach to a decision making scenario. Poker book authors have generated a lot of literature[7][8] over this approach due to its simple application and aim to help players learn the game. This method mainly works by describing common situations during a game and provides some guide lines to aid in the decision making process. The situations described using heuristics are mainly:

pre-flop hand selection: To aid in determining which hands are playable at the pre-flop stage of the game. Even though any hand may win a game, the aim of poker is to be profitable, and it will not pay off to play hands with low chances of winning a pot.

decision making: To aid in making a betting decision, whether to call, raise, fold using pot odds, dangerous cards on the table, or gaining a free card by checking.

slow playing: To aid in fooling the opponents in underestimating your hand strength by staying passive with a strong hand, or check-raising

hand reading: To aid in helping the player infer the possible hands an opponents holds by interpreting their actions.

The perceived simplicity of a heuristic based approach makes for a good starting point in building a poker playing program. The ad hoc rules in these programs use player position, betting history, hand strength (the rank of the hand within all possible hands), hand potential (the potential of making a strong hand with an upcoming board card) to generate probabilities of betting actions call, raise, fold. There are different ways of constructing such heuristics for the game, e.g by knowledge from an domain expert, or by simulated trial and error which is later refined. The problem with using a heuristic based approach is that there are too many situations to consider so the maintenance and testing of the program becomes cumbersome very quickly. Some of the research in this field includes research by Waterman[9] who attempted to learn heuristics that were represented as production rules to define a betting strategy for 5 card draw poker. His reports conclude the final strength of the program being the equivalent of that of an experienced human player. Smith[10] enhanced this research by using a genetic algorithm to learn heuristics and achieved near results using less domain knowledge. Korb and Nicholson [11] created a program that utilizes a Bayesian network mixed with simple opponent modelling to determine betting heuristics. They reported that their program can beat simple probabilistic bots and inexperienced players but loses against experienced humans and good bots. The University of Alberta Poker Research groups Poki[12] and Loki[13] bots also incorporate heuristic methods with opponent modelling to estimate hand strength and potential and are leveled at intermediate strength. The consensus reached among researchers is that an heuristic approach cannot level to a world class play[14].

Due to the nature of games of imperfect information, vital knowledge is withheld from a player at a decision nodes, which prevents the player from accurately making the correct decision. To overcome the drawback a player could assign possible hands with a probability distribution to the opponents and predict possible ways the game could play out to an ending and assess profitability based on these outcomes. This is the main idea behind simulation methods. At a certain decision point the program runs simulated games for each possible action. In each simulation run a hypothesized action is taken and the game is played out for each player. Since the simulation is subject the statistical variance many (hundred to thousands) of simulations are run and their results are averaged to calculate an EV (expected value) for each action. After the EV's are obtained for each action the program then can make a decision based on this data. The program could choose the action resulting in the highest EV.

The correctness of the EV obtained depends on the correctness of the simulations run which depend on the quality of the actions and hands assigned to opponents in each run. To generate a good EV the program must assign consistent hands and actions to the players based on their action history up to the decision point.

The simulation based methods are related to heuristic searches of game trees for games of perfect information (chess, checkers). Both methods look forward in the game tree to identify the best move. Simulation based methods traverse certain branches of the tree downward to a leaf node where heuristic searches (minimax with alpha-beta optimization) basically explore the full game tree to some predetermined depth (Fig 1).

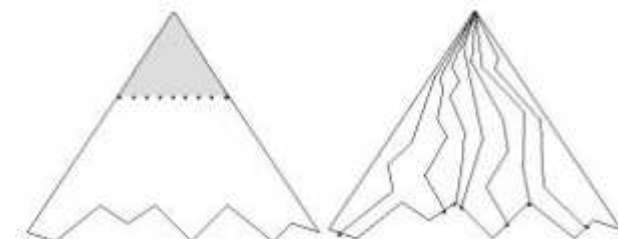


Figure 1. Minimax Game Tree search (left) Vs. Simulation (right) [13]

Simulation based methods do not deliver more than heuristics due to the difficult problem of correctly predicting opponents future moves and due to the noise and inaccuracies during simulations.

Game theoretic methods

Game theory was introduced by Von Neumann in the 1940s [1]. Von Neumann used the game poker as a basis for 2 player games and derived the first fundamental theorem the Minimax Theorem. Later John Nash incorporated results for N-player non-cooperative games. Many decision making processes can be modeled using mathematical game theory and it is used in many areas today. Poker, unlike chess or checkers is a game of incomplete information and chance outcomes, and can be represented by an imperfect information game tree with chance and decision nodes that are grouped into information sets. Since the nodes in this game tree are not independent the game tree cannot be pruned using an algorithm like alpha-beta pruning. A strategy is choosing a set of actions at each decision node. Generally this will be a randomized mixed strategy with each action chosen over a probability distribution. The player uses the same strategy across all decision nodes because from the players perspective they are indistinguishable from each other. The conventional method used to solve this problem is to convert the extensive form in to a system of linear equations are solve them using linear programming.

In the game theoretic approach to poker the game is analyzed to find a set of strategies, one for each player that form a Nash equilibrium. A set of strategies are said to be in a Nash equilibrium if no party can gain any advantage by unilaterally deviating from that strategy. Computing such a set is attractive for the following reasons:

1. The program has a known set of strategies for achieving the best possible result against its worst case opponent.

2. The programs strategy may be fully known by it's opponent and still the program will not be vulnerable to exploitation due to the nature of a Nash equilibrium.

The above properties imply that the program will break even in the worst case (against a best-response opponent) in the long run. Computing game theoreticly optimal strategies for full scale poker seems certainly intractable due to the size of the problem [2] with a 2 player Texas Hold'em game tree consisting more than 10^{18} nodes. To overcome this difficulty Takusagawa [3], Shi and Littman [4], and the University of Alberta Poker Group[5] have explored methods of abstracting the problem and obtaining approximations of optimal solutions. Dahl[6] has also explored learning game theoretic strategies for simplified poker by using reinforcement learning. One aspect worth consideration for game theoretic approaches is that they do not punish exploitable play by the opponent. A maximal player on the other hand could be made to recognize the opponents mistakes and take advantage of them. By trying to exploit an opponents weakness the program also opens its self to exploitation. Game theoretic strategies side step these risks and take a defensive stand during the game and take a risk averse approach on the profitability versus risk scale. They are built not to loose in the long run by at least breaking even.

The most successful implementation up to date using game theoretic approaches is the PsOpti program by the University of Alberta Research Group [5] which was built to play two player Texas Hold'em poker and plays at an advanced level strength. In tests against world class opposition it has held it's place for quite a long while before its weaknesses were discovered and exploited. Due to PsOpti implementing a fixed strategy once it's weakness has been discovered it can be permanently exploited. To address this problem it would need to recompute it's strategy or better yet augment it's playing by incorporating a dynamic nature that recognizes and takes counter measures against exploitable play. Research needs to be done on understanding how to create a program that is

dynamic and adapts to opponents strategies and attempts of exploitation.

Opponent modelling and prediction

A poker strategy cannot be completed without an opponent modeling system. To achieve a strong game play the model must be dynamic and adapt to the different styles and game plays of opponents. In games of complete information such a model is not essential due to the fact that playing the objectively best move will naturally exploit a player that does not respond with the best counter move. The situation with poker is different. For example one player may bluff too much in which case we should call more often and one player may not bluff enough in which case we should call less frequently. To simply call the optimal amount of times will not lead us to the most profitable play.

In poker opponent modeling is used in at least two ways.

- A general method for estimating the opponents hand strength based on their previous betting patterns
- A prediction for the next action of the opponent in the current round.

The prediction for the opponent's next action is a probability distribution of the players possible actions which is represented by the probability triple $\{\text{Pr}(\text{fold}), \text{Pr}(\text{call}), \text{Pr}(\text{raise})\}$. Guessing the next action is useful for planning advanced betting strategies such as a check-raise and is also used in simulation based methods. The challenge of opponent modeling lies in the fact that uncertainty reigns over the game. A lot of unseen cards reduce the signal to noise ratio and extracting information becomes difficult. A large number of hands need to be played even before a common situation is seen again. Missing information of a players hand at the end of each round (players may fold) prevents to model from being checked at every hand. The human psyche is a determining factor in betting decisions which is hard to model. Each player may play differently against different players. Aside from building a general opponent model for an opponent one also needs to build a model

of how the opponent models opponents in order to find exploitable holes.

Methods of prediction

Expert Systems

One way to predict an opponent's actions would be to use our own strategy or some other set of rules to make a decision. When using this type of fixed strategy we are assuming that the player will make a reasonable choice and it is referred to as "generic modeling". Although it is not a very effective method of modeling it provides a base line.

Statistics

Another method for action prediction is using the history of the opponent's previous actions as a predictor for their future actions. For example if an observation states that a player calls 40% of the time after the flop, we might reason that they will bet 40% of their hands. When the opponent's action history is used as a modeling method it is called specific opponent modeling.

Neural Networks

The University Of Alberta Poker Research Group has trained a standard feed forward neural network on contextual game data collected from online games against human opponents [13]. Their network consists of a set of 18 inputs corresponding to properties of the game such as the number of active players, texture of the board, opponent positions, etc (Table 1). The output layer consists of the fold, call and bet probabilities. After observation of the network trained on different players they concluded that certain factors are dominant in predicting opponent's actions, while others are irrelevant.

#	Type	Description
0	Real	Immediate pot odds (*)
1	Real	Bet ratio: bets/(bets+calls)
2	Bool	Committed (has put money in the pot)
3	Bool	One bet to call
4	Bool	More than one bet to call

5	Bool	Betting round == turn
6	Bool	Betting round == river
7	Bool	Last bets called by player > 0
8	Bool	Players last action was a bet (raise)
9	Real	0.1 * number of players
10	Bool	Active players is 2 (heads up)
11	Bool	Player is first to act
12	Bool	Player is last to act
13	Real	Estimated hand strength for opponent
14	Real	Estimated hand potential for opponent
15	Bool	Expert predictor says call
16	Bool	Expert predictor says raise
17	Bool	We are in the hand

Table 1. Neural Net inputs [13]

(*) *immediate pot odds are the ratio of the current bet to the pot*

Figure 2 [13] shows the network after being trained on a few hundred hands played against a particular opponent. The inputs are on the top row with activation levels ranging from 0 (fully white) to 1 (fully black). This thickness of the lines represent the magnitude of the weights (black being positive, gray being negative). In this example the connections from input node 8 (true if the opponent's last action was a bet) are very strong indicating that it is highly correlated with the next action of the player. This makes sense because if a player has just bet he is more likely to stay in the game for the next round. If their action was a check it is more likely that they will fold or check than bet. As a result of this study the player's last action was added to the context of the action frequency statistics. This improved the accuracy of the statistics by 10-to-20% on average.

The biggest drawback of neural networks is that they do not output a proper probability distribution of the likely outputs because they are not trained on a probability distribution but on the actual events. This skews the output to represent the most likely action rather than the most accurate probability distribution.

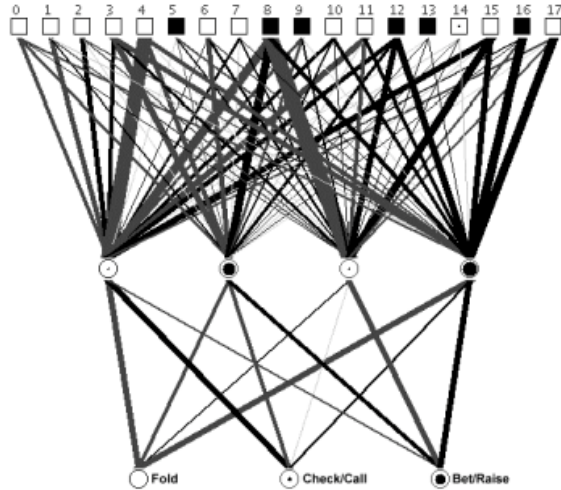


Figure 2: A neural net predicting a future action [13]

Agent evaluation

Due to the stochastic elements of the domain of poker many complex problems arise in evaluating agent performance. Observation of profitability over time is just a naïve means of evaluating performance due to the high variance of the game. Many thousands of games need to be played between opponents before a modest degree of confidence may be obtained from the collected data. Moreover the problem will increase as agent’s performances incline the number of games n to distinguish opponent performance from each other will grow at a complexity rate of $\Theta(n^2)$ [16]. Some techniques help in reducing the variance such as the duplicate tournament system, in which the same hands are played simultaneously by the agents at different positions. Since the agents do not have a memory of the game their strategies can be compared and analyzed to some degree. But in situations where agents deviate and choose different actions, the subsequent actions are no longer directly comparable. To overcome this handicap Zinkevich et al. introduced the DIVAT estimator [2006] which examines the complete history of interaction for a trial, unlike the simulation methods explained in the second part of the survey, that use a single utility of the agent’s performance per trial. As the estimator is provably unbiased (matching the players realized utility in expectation), a sample average

using the estimator provides an alternative, potentially lower variance estimate. The estimator requires a domain specific expert crafted value function that fulfills certain constraints, and the quality of the variance reduction depends on the function provided.

Conclusions

Poker is a challenging domain to computer science due to the problems explained in this survey. Research into deterministic games of perfect information has led to advances in areas of heuristic search which have found applications in various fields. Research into the domain of stochastic games of imperfect information will also lead to advances beyond the domain of the game of poker particularly because agents acting in the real world have to deal with stochasticity and imperfect information.

References

- [1] J. von Neumann and O. Morgenstern: *The theory of games and economic behaviour*. Princeton University Press, 1944.
- [2] D. Koller and A. Pfeffer. Representations and solutions for game-theoretic problems. *Artificial Intelligence*, 94(1):167–215, 1997.
- [3] K. Takusagawa. Nash Equilibrium of Texas Hold’em Poker. Undergraduate thesis, Stanford University, 2000.
- [4] J. Shi and M. Littman. Abstraction models for game theoretic poker. In *Computer Games’00*, pages 333–345. Springer-Verlag, 2001.
- [5] D. Billings, N. Burch, A. Davidson, R. Holte, J. Schaeffer, T. Schauenberg, and D. Szafron. Approximating game-theoretic optimal strategies for full-scale poker. In *International Joint Conference on Artificial Intelligence*, pages 661–675, 2003.
- [6] F.A. Dahl. A reinforcement learning algorithm applied to simplified two-player Texas Hold’em poker. In *12th European Conference on*

Machine Learning (ECML'01), pages 85–96, 2001.

[7] D. Sklansky. *The Theory of Poker*. Two Plus Two Publishing, 1992.

[8] D. Sklansky and M. Malmuth. *Hold'em Poker for Advanced Players*. Two Plus Two Publishing, 2nd edition, 1994.

[9] D. Waterman. A generalization learning technique for automating the learning of heuristics. *Artificial Intelligence*, 1:121–170, 1970.

[10] S. Smith. Flexible learning of problem solving heuristics through adaptive search. In *IJCAI*, pages 422–425, 1983.

[11] K. Korb and A. Nicholson. Bayesian poker. In *Uncertainty in Artificial Intelligence*, pages 343–350, 1999.

[12] D.Papp. Dealing with imperfect information in poker. Master's thesis, University of Alberta, 1998.

[13] A. Davidson. Opponent modeling in poker: Learning and acting in a hostile and uncertain environment. Master's thesis, University of Alberta, 2002.

[14] D. Billings, A. Davidson, J. Schaeffer, and D. Szafron. The challenge of poker. *Artificial Intelligence*, 134(1-2):201–240, 2002.

[15] L.S. Shapley, Stochastic games, *Proc. Nat. Acad. Sciences*, 39:1095-1100, 1953.

[16] D.Billings, Phd Thesis. University of Alberta, 2007.

[17] Michael Bowling Manuela Veloso. *An Analysis of Stochastic Game Theory for Multi agent Reinforcement Learning*, Carnegie Mellon University. CMU-CS-00-165

[18] Sam Ganzfried, Tuomas Sandholm. *Computing Equilibria in Multiplayer Stochastic Games of Imperfect Information*.