Poker Project - Team 47: Modelling Poker Players

AUYOK Sean, CAI Jiaxiu, CHIK Cheng Yao, Joyce YEO Shuhui, ZHUANG Yihui
National University of Singapore
sean.auyok,e0201975,chikchengyao,joyceyeo,yihui@u.nus.edu

1 Introduction

Artificial intelligence (AI) agents have long established their dominance in human games, perhaps most famously in chess with Deep Blue's win over Garry Kasparov in 1996. Even in games with imperfect information, AI agents have performed really well against top human experts. An interesting example would be Texas Hold'em Poker. In poker, a player cannot calculate all possibilities to arrive at a guaranteed optimal action, but also has to bluff and figure out whether an opponent is bluffing. More interestingly, in poker, it is not enough to win the opponent, but also to win by as large a margin as possible. This paper focuses on our team's development of a poker AI for Heads Up Limit Texas Hold'em played under strict time controls. Given the limited time for our AI agent to make an action, we discuss various optimisation strategies we adopted to improve the speed of our agent. We also discuss how we exploit imperfect information to maximise our gains in a stochastic game.

The prime functionality of our agent is that it should be rational and be able to recognise statistically optimal cards or combinations to invest in. Furthermore, our agent should be able to efficiently model the game state. In particular, we focus on how our agent abstracts card values, which is critical in processing all the different types of cards and combinations that could occur. A good abstraction allows for less calculation by our agent, which is important for the agent to play within the time controls.

Upon recognising good and bad hands, our agent has to be able to make the right decision (whether to fold, call or bet). In order to give our agent the ability to make such decisions without being predictable, we programmed our agent with Counterfactual Regret Minimisation (CFR). CFR allows our agent to quickly find Nash Equilibria and make decisions based on a probability distribution.

However, calculation alone does not make our agent a winner. We trained our agent to have the ability to model and subsequently predict how an opponent is behaving. This enhancement allows our agent to gain an advantage over randomness, and maximise our gains from our opponent's money.

1. Why did you choose this implementation (i.e. why should the reader be interested?)
2. What is the result that you achieved? (i.e. why should the reader believe you?)

2 Modelling Game State

2.1 Abstracting Card Values

In order to reduce the number of types of cards that our agent can receive, we made use of card isomorphism. Card isomorphism relies on the fact that the type of suit a card has has no inherent value and can be abstracted out. For example, A♣ A♣ has the same winning chance as A♠ A♠. Thus, by treating these cards as the same kind of hands, the agent can act similarly for both hands.

Another way we implemented abstraction was through the use of card bucketing. By grouping hands with similar hand strength, like K♡ Q♢ and Q♢ J♡, the agent can handle hands in the same bucket with the same strategy. Hands can move into different buckets depending on the community cards that have been shown. In order to get the buckets for all possible hands, we used Monte Carlo simulation to estimate the relative strength of each hand at each betting round of the poker game.

We decided on five buckets as a compromise between having enough strategies for different types of hands and keeping the game state small enough for our agent to process. Percentile bucketing was used for easy classification, where the bottom bucket corresponds to the bottom 1/5 of hands according to hand strength. In order to provide a good estimation of hand strength, we used the winrate of each hand. Each bucket is thus identified by a minimum and maximum winrate.

The buckets were generated for each betting round (preflop, flop, turn and river). For each round, 1000 different hands were generated where a hand consists of a pair of hole cards and whatever community cards are open for that betting round. For each of the hand, 500 Monte Carlo simulations were generated to determine the winrate of the hand. The bucket cutoffs were then determined from every 20 percentile of these 1000 hands' calculated winrate.

can probably put a table here

generate the win rate table by forcing both players to reveal their cards OR use a mathematical approach

cfr spne of counter-factual regret regret = heuristic

3 Modelling Opponent Behaviour

The current research on poker strategies suggests that there is no 1-king in poker tournaments. No one strategy is able to exploit all other strategies in the strategy space. The strategy graph can be described as being non-transistive, where s_1 dominating s_2 and s_2 dominating s_3 could still suggest that s_3 dominates s_1.

In that case then, an optimal agent would have to adapt to the opponent play style and find an element in the set of strategies that would dominate the opponent strategy. There are two tasks here then, firstly to abstract the strategy states for the agent, and then to approximate the opponent's strategy state based on the observed play.

3.1 Characterising Play Styles

A simple way to characterise player strategies is to consider betting behaviour when the player has bad hands and good hands.

When a player receives a bad hand, they could continue playing or choose to fold. This behaviour can be defined on a tightness-looseness scale. A tight player only plays a small percentage of their hands, and folds otherwise. On the other hand, a loose player would choose to take risks and make bets based on the potential of the hand. Loose play is called bluffing, deceives the opponent into over-estimating the agent's hand strength. The opponents may fold as a result of this observation.

Theoretically, a tight play aims to reduce losses in the case of bad hands. However, a very tight play would also mean that the chances to observe the opponent's behaviour is diminished.

When a player receives a good hand, they could call/ check (keep the pot size stable) or raise their bets. This behaviour can be defined on a passiveness-aggressiveness scale. A passive player keeps their bets low and stable. On the other hand, an aggressive player will actively make raises to increase the game stakes. Passive play is another form of deceit, which leads opponents to under-estimate the hand strength, thereby continuing to place bets and raise the pot amount.

An aggressive play style hopes to maximise the winnings when the hands are strong. However, an over-aggressive play will also encourage opponents to fold, which again discards opportunities to observe opponent behaviour.

An opponent's strategy at one point in time can then be represented as a 2-tuple of (tightness, aggressiveness). Note that this tuple merely represents an instantaneous strategy, and the opponent could change strategies over time because

1. The opponent adapts their play to our agent's play style
2. The opponent employs a team-based strategy where a coach sends out different players to the table depending on the play style our agent uses.

Heuristic for Tightness and Aggressiveness

The opponent tightness can be approximated using the folding rate over multiple games.

The drawback to using the folding-rate heuristic is that a large number of games is needed to make a prediction. By the time these observations are made, the opponent may have already adopted a different play style.

The opponent aggressiveness can be approximated using the raising rate. We make the assumption that when the opponent has a weak hand, they will only call or fold, and raising is only the result of relatively strong hands.

Our agent incorporates these heuristics into its modelling of the opponent by saving the opponent's action history each round.

Approximating Towards Opponent Strategy

Given an opponent strategy, we hypothesise that approximating our strategy to be similar to the opponent would minimise our losses. We consider extreme cases to illustrate this point.

First, consider an opponent that is very tight in play, always folding until they receive a good hand. Suppose on the other hand, that we adopt a loose play and choose to make bets even when our hands are not as ideal. Suppose then, that the opponent obtains a very good hand h_1 and we obtain a very bad hand h_2. Our looseness will lead us to contribute an amount x_1 to the pot before eventually folding or losing during the showdown. By pure chance, there is an equally probability that the two hands are reversed, so the opponents has the very bad hand h_2 and we hold on the the very good hand h_1. Since the opponent has a tighter play, they would have a lower probability of putting money into the pot, and a higher probability of folding. We let the amount contributed by the opponent to the pot be x_2. Clearly, x_2, the expected loss by the tight opponent ¡ x_1, the expected loss by our loose agent. In the case of tightness, it then makes sense to approximate our tightness to be close to that of the opponent.

Alternatively, consider an opponent that is very aggressive in play, always raising when they receive a good hand. Suppose on the other hand that we adopt a passive play and choose to call or check even when our hands are very strong. (expound on this)

Randomising Strategy

The exploitability of opponent behaviour should be symmetric, so our agent has to acknowledge the opponent can similar extract patterns from our play style. It would then be beneficial to either mask our play style or change our play style during the game.

Our agent model generates a 3-tuple for the probability of playing each action (fold, call, raise). For example, one of the tuples generated could be (0.2, 0.3, 0.5). We note here that the probability of raising is the most significant, so we can deduce that our player hand is strong.

Some researchers employ purification techniques to overcome abstraction coarseness. These poker agents prefer the higher-probability actions and ignore actions that are unlikely to be played. In particular, Ganzfried and Sandholm found full purification to be the most effective. The full purification technique let the agent play the most-probable action with probability 1.

However, such a strategy ... makes it predictable?? In particular, the deceit techniques of bluffing and trapping would be less effective.

3.2 Predicting Play Styles

What game states can we use to approximate play styles? Tightness - Folding Rate Aggressiveness - Raising Rate

4 Training

dqn

Acknowledgments

A LaTeX and Word Style Files

References