CS-512

ARTIFICIAL INTELLIGENCE

TERM PROJECT

INDIAN INSTITUTE OF TECHNOLOGY ROPAR, PUNJAB

A New Smart Opponent in Poker

Team Members : · Anurag Jaiswal (2021CSM1002) · Arunava Chaudhuri (2021CSM1020)

May 8, 2022

1. Introduction

As we know that poker is generally considered as a classical problem which is a member of imperfect information games in modern game theory. It instinctively and exquisitely captures the challenges of latent information for every individual opponent. We have seen that time complexity of this type of problem-solving method is generally higher with respect to perfect information games. The game of poker for its unique characteristics and different strategies have been researched for years. Scientists and analysts of AI are devoting their time to find its problem-solving method just as in Alpha Zero or AlphaGo. Nonetheless, Texas Hold'em poker consists of further risk management, misleading deceptions, dynamic decision-making, and, as well as multi-stage chip, challenges of imperfect information etc., that restrain it from being explained entirely by AI. Although many scientists have strongly assured that the associated technique behind Texas Hold'em Poker's solution can be broadened to many real-world applications, such as military applications, cybersecurity, finance, auction and strategic portfolio, and the encouraging application anticipation inspires extended research until now.

In general, this type of poker is a collective decision game containing of four steps: preflop, flop, turn, and river. At each stage, a poker players can either bet dissimilar quantities of money based on own hands and public cards. The players can only acquire rewards after taking into consideration a series of sequential actions until only one player remains in or at the end of the last river stage. Depending to the restriction of amount for betting, this game can be separated into either a limited game or a no-limit game. As it is given that number of their information sets is about 10^{14} and 10^{162}, respectively. Now, Solving Texas Hold'em (TH) is much more complicated and consume, which converts TH a valuable benchmark in the domain of higher-scale imperfect information game play.

Traditional agents in poker like games calculate an ideal technique by resembling a Nash Equilibrium using a minimax tree search by using nodes of feasible hands and its actions and executing an evaluation function to find out the benefit of every node. Now the next task is to determine the robustness of the player's hand at each node, which is calculated by estimating the viable future game states. We know that poker is of stochastic nature, that is why its branching characteristic of its states is of large-scale and it creates an enormous search time and search space. So, here, inn this paper we have aimed to do the following:

1. We will define a hand strength function to calculate effective hand strength
2. We also define some probabilistic function for predicting actions of player in each round
3. We will use hand potential function to estimate how much the player is ahead before the next round

The final evaluation will be done using Monte Carlo simulator to check winning percentage of player during the game.

2. Motivation

For years, many researches have been conducted in the area of imperfect information games and most of them was try to solve the problem of creating an agent who can beat an intelligent human poker player who is capable enough to compete in the popular version of poker that is Texas Hold'em. Here the agent must have the capability to manage risk, make a decision to optimal it's strategy in real world scenarios and have the ability to create a realistic assumption about opponent moves along with adaptability to work in a situation where only deception and unreliable information is available. Like other games such as chess where a strategy does not rely on opponents and the player can play his best reply with an assumption the opponent plays rationally. But in game of poker the strategy is to make use of the action of weak rival with respect to the variant if mistakes every rival makes.

3. Related Work

Until now many works have been done for card game like Poker or UNO. For poker there is many variants of popular competition Texas Hold'em like Limit Hold'em, No Limit Hold'em, Leduc Hold'em etc. Researcher have been trying to create a single agent which can optimal in all of these poker game type. But as of now there is several solutions for individual poker variant. Fredrik A. Dahl's "A Reinforcement learning Algorithm Applied to Simplified Two-Player Texas Hold'em Poker" (1) talks about an approach of implementing reinforcement learning for modelling an agent in poker. The end result is somewhat towards increasing the rewards for the agent compared to naïve approach. G Nicolai and RJ Hilderman's "No-Limit Texas Hold'em Poker agents created with evolutionary neural networks" (2) has proposed an innovative approach for building the logic for each action for poker agent in a testbed of Monte Carlo simulation. The result of this approach is also close to the reinforcement learning based approach but, in some situation, it can have better improvement compared to previous one. The first paper on "Opponent Modelling in Poker" (3) by Darse Billings, Denis Papp, Jonathan Schaeffer, Duane Szafron first approach towards adopting the opponent's mistake and learn the agent to utilize those opportunities. In this paper we have seen the first use of evaluating hand strength for a pair of cards that a player can hold. Based on that rank that player's possibility of winning the game. They had also introduced some potential value which was used to calculate effective hand strength of player. Also, the paper "Bayes' Bluff: Opponent Modelling in Poker" (4) by Finnegan Southey, Michael Bowling, Bryce Larson, Carmelo Piccione, Neil Burch, Darse Billings, Chris Rayner suggested to use Bayesian probabilistic model to predict the action of opponent in a particular round. These two are the techniques which is different from some conventional approach in this problem. So, we have planned to combining the Bayesian probabilistic model and hand potential, hand strength function for effectively building the opponent model in poker.

4. Method

We have proposed a basic model for modelling the agent in this poker. It also describes how the process generates an action in a open game state. At first the information about the public game state will be sent to the hand evaluator function and the opponent modeller block. Hand evaluator will have the input from betting rule base. Now the information of hand card of the current player is also sent to the faction. After evaluation the result is send to opponent modeller block. Then the output of opponent modeller will feed to the opponent model block. Here the model will check the result of opponent modeller against income value of each hand combination and based on that information it generates some potential values and send it to the hand evaluator block again. Now the Hand evaluator function will use the Bayesian probabilistic method to predict the action based on the potential value and effective hand strength. After that the information will go to the player and based on that the player will play his move in that game round of poker.

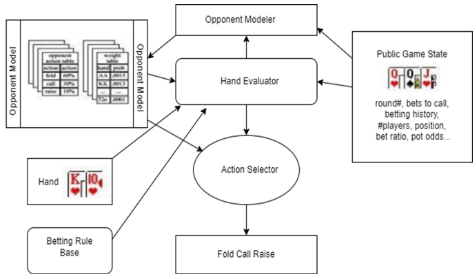

Figure 1: Basic Model of Poker Playing Program

The game poker has several round as we have discussed previously. Now depending on that we have divided out methods for opponent modelling in some parts.

1. Evaluation of Preflop round.
2. Hand Potential and Hand Strength
3. Betting Strategy
4. Modelling the Opponent

Now we will discuss about each of these steps in the following section.

4.1. Evaluation of Preflop round:

In Texas Hol'em Preflop round has been researched broadly for many years. Those efforts are illustrated as an explanation towards understanding the terms in human specific manner. This evaluation process categorized the two cards' pair in preflop round. Now for each category of hand card a betting action is suggested and this technique will work as specialist system but it will work more systematized manner and will have the adaptability to modify and generalize its ideas after each round of play.

At first, starting of the game, two cards are dealt to each player playing the game and there is no private card on the table. So, in this way the total number of possible pair of card combination will be { 52 *choose* 2 } which is equal to 1326. Out of that combination a player can hold one of them but all the card combination are not distinct while considering the potential prior to the public cards are to be shown. Now depending on this criterion, the number of combinations will decrease to { 13 choose 2 } *2 +13 which is equal to 169 different hand pairs. Now of those 169 hand pairs a simulation of 10 lakhs poker games had been done at odds with nine random hands (Fig. 2).The result generates a statistical measure of estimated income rate or profit expectation for every initial hand. The result shows that the highest income rate can be obtain by a player if he has a pair of aces, whereas the lowest hand will be a pair consisting of cards numbered in range from 2 to 7 of various suit.

Opponents:	1	2	3	4	5	6	7	8	9
AA	85.3	73.4	63.9	55.9	49.2	43.6	38.8	34.7	31.1
KK	82.4	68.9	58.2	49.8	43.0	37.5	32.9	29.2	26.1
QQ	79.9	64.9	53.5	44.7	37.9	32.5	28.3	24.9	22.2
Aks	67.0	50.7	41.4	35.4	31.1	27.7	25.0	22.7	20.7
AQs	66.1	49.4	39.9	33.7	29.4	26.0	23.3	21.1	19.3
JJ	77.5	61.2	49.2	40.3	33.6	28.5	24.6	21.6	19.3
KQs	63.4	47.1	38.2	32.5	28.3	25.1	22.5	20.4	18.6
AJs	65.4	48.2	38.5	32.2	27.8	24.5	22.0	19.9	18.1
KJs	62.6	45.9	36.8	31.1	26.9	23.8	21.3	19.3	17.6
ATs	64.7	47.1	37.2	31.0	26.7	23.5	21.0	18.9	17.3
AKo	65.4	48.2	38.6	32.4	27.9	24.4	21.6	19.2	17.2
TT	75.1	57.7	45.2	36.4	30.0	25.3	21.8	19.2	17.2
QJs	60.3	44.1	35.6	30.1	26.1	23.0	20.7	18.7	17.1
KTs	61.9	44.9	35.7	29.9	25.8	22.8	20.4	18.5	16.9
QTs	59.5	43.1	34.6	29.1	25.2	22.3	19.9	18.1	16.6
JTs	57.5	41.9	33.8	28.5	24.7	21.9	19.7	17.9	16.5
99	72.1	53.5	41.1	32.6	26.6	22.4	19.4	17.2	15.6
AQo	64.5	46.8	36.9	30.4	25.9	22.5	19.7	17.5	15.5
A9s	63.0	44.8	34.6	28.4	24.2	21.1	18.8	16.9	15.4
KQo	61.4	44.4	35.2	29.3	25.1	21.8	19.1	16.9	15.1

Figure 2: Winning percentage for various hands

4.2. Hand Potential and Hand Strength:

After completion of Preflop round, three public cards will be revelled. In each round of flop, turn and river it is crucial to evaluate the strength of hand pair card as the it tells how much ahead or behind the player's possibility to win that round. Also using enumeration technique, we can calculate the probability of holding the best pair of cards at any the time when three table cards are disclosed. Here this probability value is defined as "Hand Strength". In naïve approach the player fully depends on this probability value but in our case we this approach will not work. The reason is as the next two rounds come; next two cards will be disclosed. In that situation the potential of being ahead or behind will change. Without the concept of opponent modelling the player will count the number of possible best or worst hand card pairs compared to its own cards pair. In general Hand Strength function is inadequate to process the standard of a hand card pair. Let say, a player has a card pair as 5-Hearts and 2-Hearts and on table there are three cards such as 2-Hearts, 4-Clubs and J-Hearts. As per poker this combination is weak hand pair however tremendous potential for enhancing its hand card. Other than that, out of next two cards that yet have to be disclosed, if one of them become any hearts, or Ace or 6 will generate a straight or flush type 5 card set. So, the player needs to be attentive enough to sense the probability of improving to the better-quality hand when the player is behind. That is why we have defined two more factors which are described below. It is called as Hand Potential value. It has two types such as:

1. **Positive Potential (Ppot):** It gives the value for probability of enhancing the player's strategy when the player is behind some of the opponent in the game.
2. **Negative Potential (Npot):** It gives the value for probability of lagging behind when the player is ahead.

As we have possibility of 1081 opposing hand and out of those, we will consider 990 card pairs for the next two public card on the table. We will calculate for every above card pair in how many situations we will fall behind or tied or ahead. We have also shown an example of card pair where in hand there was cards like A-Diamonds, Q-Clubs and the public cards 3-Hearts, 4-Clubs, J-Hearts against a single rival in fig. 3. Here the rows are defined with the final standing on the flop round. The columns are defined as the final standing after the last two public cards are disclosed. In Figure 3, we have measured the potential depending on two additional cards. This method is known as two-card lookahead and it generates a Npot2 of 0.274 and a Ppot2 of 0.208. We also measure same computation depending on one-card lookahead (Ppot1) which have only 45 probable next cards (44 if the player on the turn) rather than 990 end results. Corresponding to one-card lookahead on the flop, Npot1 is 0.145 and Ppot1 is 0.108.

5 Cards	7 Cards			
	Ahead	Tied	Behind	Sum
Ahead	449005	3211	169504	621720 = 628x990
Tied	0	8370	540	8910 = 9x990
Behind	91981	1036	346543	439560 = 444x990
Sum	540986	12617	516587	1070190 = 1081x990

Figure 3: A-Diamonds, Q-Clubs / 3-Hearts, 4-Clubs, J-Hearts potential

Above measurements which comes up with exact probabilities which judges each probable situation into consideration, giving smooth, vigorous results. Although, the approximation that all card pair rival cards are equally likely is false, and the measurements must be altered to mirror this.

4.3. Betting Strategy:

Now the time it is the player's turn to act he will measure hand strength and hand potential to choose a betting move. Now after all those value collection we will measure the value for Effective Hand Strength (EHS) which is calculated as below:

$$\text{EHS} = HS_n * (1 - \text{Npot}) + (1 - HS_n) * \text{Ppot} \quad (1)$$

We have also measured the value for Pot Odds which depicts the player's winning chance with respect to the expected recovery from the pot. Let's say in a betting round size of pot is p, size of bet is b which is the money to be bet to stay alive in the game. Then we can measure Pot Odds as:

$$\text{PotOdds} = b/(b + p) \quad (2)$$

After that we defined another variable d which is calculated using the following function:

$$D = \text{EHS} - b/(b + p) \quad (3)$$

In general case for naïve approach the player sets and deterministic boundary for on d for call, fold and raise move. But that technique can be capitalized by rival and the player cannot do anything in case of sandbagging and bluffing. In that we will use Bayesian Probabilistic approach for betting which is defined below:

$$P(\text{bet}) = 1/(1 + e^{-a(d-f_1)}) \quad (4)$$

$$P(\text{fold}) = 1/(1 + e^{a(d+f_2)}) \quad (5)$$

$$P(\text{Call}) = e^{(-20((d+f_c)^2))} \quad (6)$$

Here the values of those constants can be varied depending on the characteristic of rival which is defined by the history of hands of rival. We have plotted the action probability curve (Fig. 4) against each of the pot odds value to visualize the action taken by the player at those pot odds values.

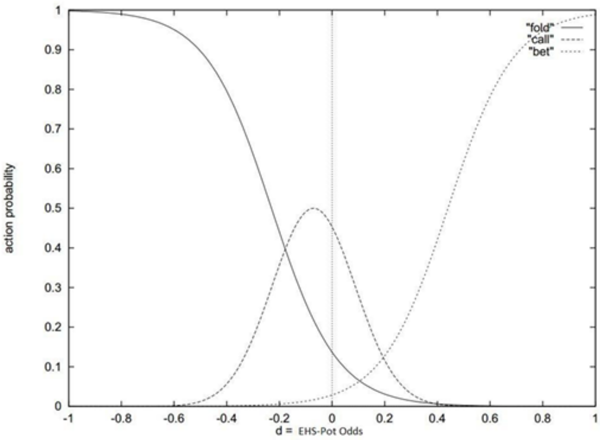

Figure 4: Action Probability Curve.

4.4. Modelling the Opponent:

Many strategic games like chess, we have not seen any type of loss in performance without using the model referring the opponent strategy, hence it is neglected. In case poker this opponent model task has an extensive impact on a player's chance of winning and works as a differentiating characteristic among the players playing poker with distinct skill set. Now suppose a group of players have all the knowledge of poker foundations and have the capability to change decision depending on an exact model of the rival which have a higher effect on winning than any other key method.

There has been some discussion about opponent modelling which is fundamental for imperfect information game like poker where the actual technique of collecting data and utilize it for betting method is a complicated and fascinating problem. For our problem we have divided opponent modelling as a three steps process which is discussed below.

4.4.1 Weighted Enumeration

The above-mentioned naïve approach, we have approximated that opponent is holding every probable card pair with equal probability. But this technique will not work in case the player has cards which has weak potential, in that situation the player would have folded and quit the game in earlier round. The solution for this problem is to use the weight which defined the probability of rival got that card which the player has not folded yet and it is also given that the game is open prior to the moves used by both players. Now when a player takes any actions the weight value will also change.

4.4.2 Initial Weight Calculation:

Here we have measured the initial weight values depending on the history of the game and rival's moves. We will compute the median and variance of rate of income for calling, raising and folding hand. After that we will use linear interpolation around the median value along with median which have 0.5 as weight value.

4.4.3 Reweighting

Every time a rival makes a betting move, the player will alter the weight values by introducing a transformation function. As a simple approach we have not used re-weighting in Preflop round. But from the next round likewise, Flop round, we have defined a variance and mean value which works as a threshold for the rival's noticed move. As it is impossible to map variance and mean to the list of all ranked card pair, rather than we have ranked all of the possible five set cards which are created from each initial hand cards and three public cards. We have measured EHS to rank those five-card set.

Now for every five-card hand, the calculated re-weighting factor is multiplied by starting weight to generate the modified weight. This step is replicated for every noticed betting action all along the game. At the last round of betting, a particular rival may have only a smaller number of hands which has comparatively high weights which means that the technique become successful to narrow down the search space for the possible hand combination. An example of possible hand reweighting calculation is shown in figure 5.

5. Experiment

To implement our own method, we have built a code for poker game with all of its essential function from scratch. For convenience, we have set the number of players to six at the starting of the preflop round. Also, initial pot value is set to 20$. In the utility functions we have implemented two potential functions like Npot and Ppot with card ranking function. We have also used a predetermined file of data related to different hand card pair with its corresponding winning percentage for six number of players. After that some constant terms is also defined in file. A constant term flushes is used for finding all flushes hand and straight flushes hand. There is some entry in the matrix which is set to zero, denotes that the combination is not possible with a five-card flush hand. Another lookup table is also created for all non-flush hands which is of five-card set. It denotes a set of five card is either Straights or High card hand. We have also set thirteen prime numbers for thirteen distinct card type here to identify each distinct card in poker. Some possible combinations value for card pair is calculated beforehand to reduce calculation time in our code. Hand strength, effective hand strength, card rank this kind of function are also defined in our code. For evaluation purpose we have created a separate function to calculate winning percentage for different group of players starting from 2 to 10. We have also plotted that data corresponding to each iteration in our experiment. After each round of game this code will show the player win the game along with the five-card set for which is declared winner. These data of five card along with the hand card of the player will go to a simulator. We have used a Monte Carlo simulator which basically tells us the winning percentage of that card hand with those cards on the table. Also gives us an idea about losing percentage and percentage of game being tied. It has a better view about the winning hand card type along with percentage of winning using those hands corresponding to the losing hand and its percentage of losing against the winning hand.

Hand	Weight	HR	HS_1	$\sim PP_2$	EHS	Rwt	Nwt	Comment
J♣ 4♥	0.01	0.993	0.990	0.04	0.99	1.00	0.01	very strong, but unlikely
A♣ J♣	1.00	0.956	0.931	0.09	0.94	1.00	1.00	strong, very likely
5♥ 2♥	0.20	0.004	0.001	0.35	0.91	1.00	0.20	weak, but very high potential
6♠ 5♠	0.60	0.026	0.006	0.21	0.76	0.90	0.54	weak, good potential
5♠ 5♥	0.70	0.816	0.736	0.04	0.74	0.85	0.60	moderate, low potential
5♠ 3♠	0.40	0.648	0.671	0.10	0.70	0.75	0.30	mediocre, moderate potential
A♣ Q♦	1.00	0.585	0.584	0.11	0.64	0.60	0.60	mediocre, moderate potential
7♠ 5♠	0.60	0.052	0.012	0.12	0.48	0.20	0.12	weak, moderate potential
Q♠ T♠	0.90	0.359	0.189	0.07	0.22	0.01	0.01	weak, little potential

Figure 5: Re-weighting various hands after a 3—Hearts, 4Clubs, J-Hearts flop (m = 0.6, s = 0.2).

It can tell us about the pot value growth after full round of play in poker.

6. Result

After the experimental setup we have successfully ran our code in Spyder. The result shows us that in each round of play the AI players is changing its decision for selecting its moves. At the starting of the first few game the chances of the agent's winning is less than that of aggressive player or the efficient human player. But as time goes by, the agent will learn from the mistakes and actions taken by its opponent in this game. We have run the simulator at the end of each gaming round for the agent in case of winning or losing. At the starting the winning hand percentage is below 20% in all the cases but after some iteration of play the agent's wining percentage increases as we have seen in simulator. We have seen some strange behaviour whenever an expert human player plays against the agent. At starting the agent will lose but after some time in some cases it beats the opponent expert player but not all the time. We know that the method we are running here will take some time to adopt the strategies implemented by expert player. From the betting curve (Fig. 4) we can assume that at negative pot value there is high chance that the agent will go for fold move. After that as the pot value increases the probability value increases to go for call move and at the end for high value of pot it surely wants to do a bet in a round.

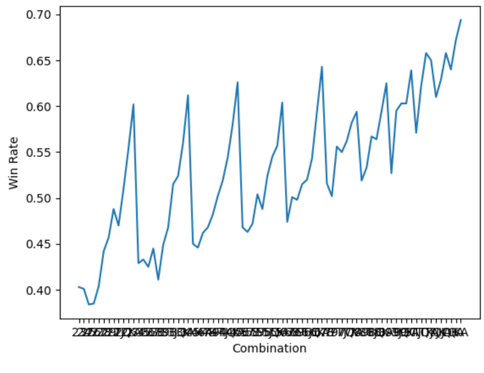

Figure 6: 2 Players (Suit) winning percentage for each hand combination.

Next evaluation is on wining percentage against a group of players and with distinct hand card pair. Using different programming environment, we have seen that initially for two player game the winning percentage is high. As the card pair combination becomes more and more higher ranked the percentage of winning of that high ranked hand increases. What we can see from Fig. 6, that the winning rate can goes all the way to maximum of 70% using 2 player suits. This scenario changes slowly as the number of players playing the game increases. We see that when the players number increases the chances of inning using a particular hand combination decrease compared to 2 player suits.

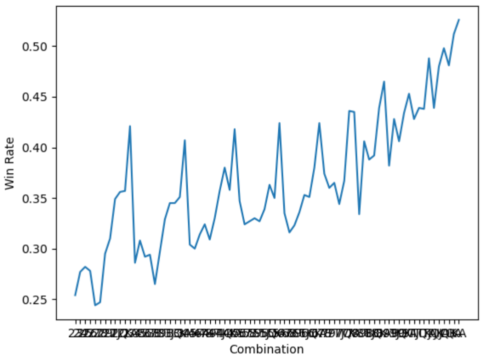

Figure 7: 3 Players (Suit) winning percentage for each hand combination.

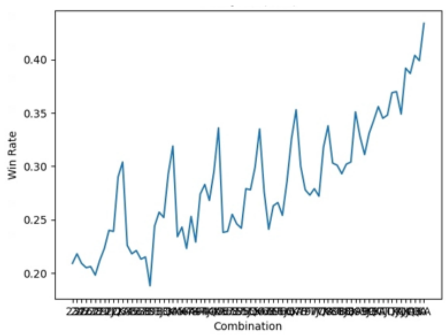

Figure 8: 4 Players (Suit) winning percentage for each hand combination.

From Fig. 7, it is evident that the chances are dropping for all of card's combination. This is the real-world situation for a game like poker where there is various probability of winning for a particular player. As an imperfect information gaming test-bed it gives us a better idea how the agent should develop itself to compete against a human player. Sometimes it may happen that it looks like using 3, 4, 5 or may be 6 player suits have lower probability of winning but logically if an agent has the ability to learn from opponent's mistake it may have the ability to win every time and its chances will always be in the small percentage winning rate as we can see from the graphs. We have also noticed that the graph lines are fluctuating for all of the hand card types.

The reason we assume that at each hand card types if the player think strategically that the card pair is not suitable enough to give him a winning bet, he will try to flop that card and quits that game in that case the winning percentage may be increases or decreases. We have also noticed that the graph lines are fluctuating for all of the hand card types.

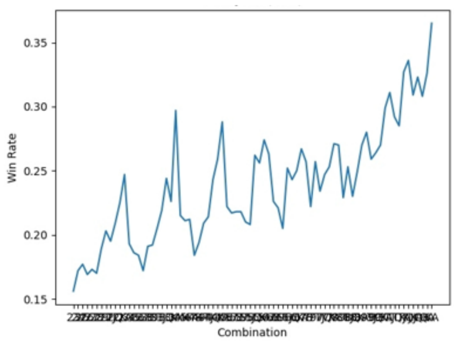

Figure 9: 5 Players (Suit) winning percentage for each hand combination.

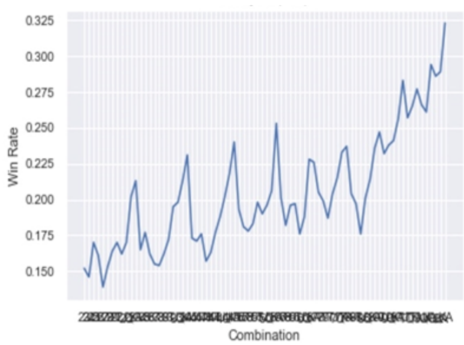

Figure 10: 6 Players (Suit) winning percentage for each hand combination.

7. Conclusion & Future Work

In this project, we have created a poker game environment from scratch which is inspired from Texas Hold'Em Poker Technique. In which one player is playing using the probabilistic model technique to improve its betting strategy against other randomized and a human player. A good and well opponent modelling against Human is still a challenge. In above works when opponent strategy doesn't vary much with time. Here number of games required for above specified strategy to learn modelling is not of practical use as we don't get same opponent to play that much no. of games. Bayesian network can be used to model uncertainty in game as well as opponent. Although using this strategy has produced some of the best result compared to other other discussed above. In the future we have model the probabilistic method in a more logical manner so that it will eventually learn from the opponent mistake more efficiently and can able to defeat a human poker champion player in Texas Hold'em Poker competition.

References

A. Dahl, Fredrik A, "A reinforcement learning algorithm applied to simplified two-player texas hold'em poker," in *European Conference on Machine Learning*, pp. 85–96, Springer, 2001.

G. Nicolai and B. Hilderman, Robert J, "No-limit texas hold'em poker agents created with evolutionary neural networks," in *2009 IEEE Symposium on Computational Intelligence and Games*, pp. 125–131, Ieee, 2009.

D. Billings, D. Papp, J. Schaeffer, and D. Szafron, "Opponent modeling in poker," *Aaai/iaai*, vol. 493, no. 499, p. 105, 1998.

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