Bayesian Opponent Modeling in a Simple Poker Environment

R.J.S.Baker and P.I.Cowling
MOSAIC Research Centre, Department of Computing,
University of Bradford
Bradford, BD7 1DP, UK.
E-mail: R.J.S.Baker and P.I.Cowling@bradford.ac.uk

Abstract— **In this paper, we use a simple poker game to investigate Bayesian opponent modeling. Opponents are defined in four distinctive styles, and tactics are developed which defeat each of the respective styles. By analyzing the past actions of each opponent, and comparing to action related probabilities, the most challenging opponent is identified, and the strategy employed is one that aims to counter that player. The opponent modeling player plays well against non-reactive player styles, and also performs well when compared to a player that knows the exact styles of each opponent in advance.**

Keywords: Poker, Opponent Modeling, Bayesian Analysis

I. Introduction

Poker is a card game believed to have existed since the sixteenth century in Germany as a bluffing game called "Pochen" [1]. It developed in the 1830's into the game now known as Poker and since has been a relatively popular game in casinos and private houses. During recent years, poker has seen a large popularity boost, partially due to regularly televised tournaments, and the availability of online play, despite recent problems due to new US legislation banning banks from processing internet gambling transactions [2].

The basis behind all games of poker is relatively simple; each player is dealt a set number of cards (varies for different styles of poker, usually between one and five), and then a betting round commences, within which each player tries to convince their opponent that they have the best hand; the strength of the hand being inversely related to the probability of receiving that set of cards. There can be few or many of these betting rounds, in the case of Texas Hold 'Em, there are 4 rounds (pre-flop, post-flop, turn, river), but other forms vary. Three actions can be taken in the game which apply to all forms of poker, these are

1. ***Bet/Raise***: Add money to the pot, and increase the monetary risk for the player and the opponents
2. ***Check/Call***: Take no action if the player's current bet is equal to the minimum required bet, or put in a bet equal to the current bet.
3. ***Fold***: Take no further part in the proceedings of the hand.

These basic actions are an essential staple of all poker games. It is the general uncertainty of not knowing if the opponent's actions represent a truth or a lie that makes the game of poker one of the most skilful card games in the world.

Poker has driven numerous research efforts, although less so than some other games, arguably due to the difficulty in modeling with imperfect information. Games such as chess have a fully available state of information meaning that every possible move can be predicted and modelled, spurring many research developments, most famously being the interest and controversy surrounding Deep Blue [3]. Poker is different in the sense that the only piece of information held by a player of the game's state is that of their card(s) held, and that of any past actions the opponents have made. This paper investigates the creation of a player for a simplified form of poker, which analyses and reacts to opponents' play.

The scientific investigation of poker has always been an extremely complex field, and with the number of possibilities that each hand can hold, there is no real analytical way to cover all possibilities in a suitable timeframe. Koller [4] states that standard techniques (such as a minimax search algorithm) are not reliable for accurately determining the reasoning required due to the prospective use of bluffing; an opponent's bet could represent a bluffing move, or a show of confidence, this being difficult for any player to determine. Koller then explains the *Gala* system, which uses a language similar to Prolog to describe and solve imperfect information games such as poker, using a three-card deck with one card dealt per player. Several approaches to understanding the mechanics of imperfect information games have been based upon simplified variants of poker: such as [5], where three simple versions of poker are used to investigate recommended changes in players' actions dependent upon the number of opponents. Billings [6] has considered means of reducing the complexity of the gaming situation, reducing betting rounds by defining 19 possible sequences that each round can take. The paper also covers the elimination of multiple betting rounds, to simplify the problem somewhat, but maintaining the fundamental nature of poker. Billings also created a statistics-based opponent modeling system, where each opponent action is used to generate a simple probability of each action, and the player uses the probability to guess the opponent's next action [7].

The modeling system uses a neural network based system that, when given a set of inputs of an opponent's previous actions and the current state of play, will produce a probability distribution of the opponent's next action. Schaeffer [8] identified and defined some ground requirements for creating a 'world class poker player', which are not independent, but altogether form an integral part of the *Loki* system. These requirements are Hand Strength, Hand Potential, Betting Strategy, Bluffing, Unpredictability and Opponent Modeling. One other rule that is inferred from the paper, but not explicitly cast as a rule, is that of adaptability; a good player who analyses the opponent's playing style would easily exploit a deterministic means of decision making.

Opponent modeling has been seen as having a greater impact on success in games of poker than most other games, indeed poker may be seen as a testbed for opponent modeling research In the case of [9], modeling is implemented by adjusting weights in relation to actions made and the success of the opponent as a result of those actions. Eric Saund explained an approach that captures and analyses betting actions in relation to inferring the downcards held by an opponent in seven-card stud poker [10]. Saund uses several game situations, one in which all cards are viewable, and one in which some cards remain hidden. Despite having card information that would be unseen in a real life game, Saund shows that using betting actions to ascertain how an opponent will play is much better than approaches that do not use such information.

II. One Card Poker

We use a simple version of poker, which still maintains some useful parts of the 'flavour' of a full-scale poker game. The deck consists of ten cards, labeled from 2 to A (Ace being the strongest card, but names and suit are arbitrary, only the strength order of the cards is important). The game consists of four players; each player has an initial credit of 10 chips, and each hand entered requires a one-chip ante from each player, after which each player is dealt one card, similar to the approaches of [11], which uses an 8-card deck with each player having only one card and one chip each, and [12] in which each player is dealt a card classed as 'high' or 'low', but can reach the situation of a 'draw' where each player holds a card of the same strength as it's opponent and the pot is shared. The winner of the hand is the player with the highest valued card; the initial difference between any other variation of poker and this simplified one is that of the case where a player holds an Ace; the player knows that the contents of the pot are theirs as no card can beat an ace implying a definite win. This means that damage limitation must be employed by players to make sure that an opponent with an Ace does not fleece the other players of all their chips.

After the cards are dealt, the players take turns in a clockwise direction, and make a decision whether to bet, fold, or check, given the value of their card. Betting (which is equivalent to a 'raise' action), and each subsequent raise costs one chip. Once all players have matched one another's bets, or all but one player has folded, the showdown is reached, and the player with the highest card (or only player remaining) receives the pot contents. The players continue playing further hands until there exists a tournament winner who possesses all of the chips. The game itself has no maximum bet; so entire tournaments could theoretically end in one hand, dependent upon the play style of the players.

III. Experimental Design

A. Design of the Distinct Style Players

Poker players may usefully be categorized into four main styles, these are explained in [13] to be:

- ***Loose Aggressive (LA)***: A loose aggressive player is one that is loose (over values card(s) held, and even stays in hands it is unlikely to win) and is also aggressive (when given the opportunity to stay in the hand, will bet or constantly re-raise forcing the pot higher).
- ***Loose Passive (LP)***; A loose passive player will also over value their cards, but take very little action apart from checking/calling, and will only bet in a rare situations in which they will believe that winning has a very high probability.
- ***Tight Aggressive (TA)***: A tight aggressive player accurately values their card in terms of win probability, and will more often fold (which is mainly attributed to tightness), but due to the player's aggressive nature, any hand that it is determined that the player should stay in, then the player will bet to force the pot value higher.
- ***Tight Passive (TP)***: A tight passive player plays very few hands, and even when doing so will very rarely make a betting action.

Each of these styles of player was modeled using a simple deterministic design [fig 1]. A player's style is characterized by the probability pair [α, β], where α represents the minimum win probability required for a player to remain in the hand, and β represents the minimum win probability for the player to bet. It should be appreciated that α is responsible for whether a player is tight or loose, and β for whether a player is passive or aggressive. If the win probability is less than α, the player will make a checking action if there is no additional monetary risk (i.e. no money needs to be placed in the pot to remain in the hand), and fold otherwise.

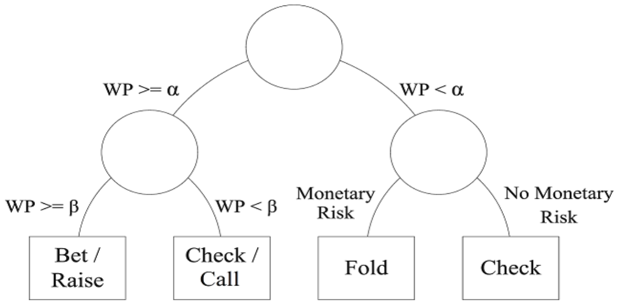

Fig. 1. Layout of a general player

A pair [α, β] represents a deterministic player, with a distinct play style. The α and β values for each playing style is defined in Table I.

TABLE I
α AND β VALUES FOR EACH STYLE OF DETERMINISTIC PLAYER

	α	β
LA	0.1	0.2
LP	0.1	0.9
TA	0.5	0.6
TP	0.5	0.9

B. Design of the Anti-Players

"Anti-Players" were created as a 'nemesis' to each of the LA, LP, TA, and TP players. The values of α and β for an "Anti" Player are dependent upon the number of opponents, and the respective styles of those opponents. [α, β] pairs were tested in increments for $0 \leq \alpha \leq \beta \leq 1$ to determine the best value. Each [α, β] pair was tested in a 100-game four-player tournament, with all players starting each tournament with 10 chips. For example, fig. 6 gives the performance of different [α, β] pairs against three LA players. Table II gives the [α, β] values for the Anti-LA, Anti-LP, Anti-TA, and Anti-TP players, with the success rate of these values against the four distinct styles. Against loose players, the α and β values represent a rationally tight style of play, as loose players will often squander chips when facing a tight opponent that holds a strong card. When playing against tight players however, a loose strategy is adopted to remaining in play, and a tight one in relation to betting/raising. This appears rational, as many tight players will fold when holding a weak card, possibly leaving the pot to a looser player that may hold a weaker card. The tightness in relation to betting is also rational; staying in hands where the pot is small is wise, but when a strong card is held, the player should try to raise the pot as high as possible. When against loose players, the win percentage cannot reach 100% due to the situation where the loose player has the highest card and each player bets, ultimately costing the Anti Player all of its chips. Conversely, when facing tight players, the win percentage cannot reach 100% due to the Anti Player remaining in all hands, meaning that in some circumstances, the player will run out of chips due to the ante per hand. It should be noted, however, that the usage of tight and passive play is somewhat exaggerated.

TABLE II
α AND β VALUES FOR EACH STYLE OF ANTI-PLAYER

	α	β	WIN %
ANTI - LA	0.6	0.8	76
ANTI - LP	0.8	0.9	63
ANTI - TA	0.0	0.7	71
ANTI - TP	0.0	0.8	75

C. Modeling the Opponent with the Analysis Player

Having a means of defeating each style of player raises the question as to whether we can combine them to create a player that could defeat all types and combinations of opponent. This player, the Analysis Anti-Player would need to use the past history of each player's betting actions to determine the style of each opponent. Bayes' theorem could be used to analyse the past play information of an opponent, and aims to determine the style of each opponent The usage of Bayesian probabilities to model uncertainties has become popular in relation to imperfect information games, such as Poker [15].

Bayes' theorem relates conditional and marginal probability distributions of random variables, which shows that however different the conditional probability of event A conditional upon event B is to that of B conditional upon A, there is still a relationship between the two.

$$\Pr(A \mid B) = \frac{\Pr(B \mid A)\Pr(A)}{\Pr(B)} = \frac{\Pr(B \mid A)\Pr(A)}{\sum_{a} \Pr(B \mid a)\Pr(a)} \quad (1)$$

The Analysis Anti-Player uses Bayes' theorem to calculate the probability of a player utilizing a specific play style. Equation (1) gives Bayes' theorem where A is a random variable representing player type, and B is a random variable representing player action. Pseudocode for this analysis player is given in Fig. 2.

```
for All of Opponent i's past moves
{
  action = Players[i].PastMove;
  // Multiply each action probability by the initial
  // probability
  tp = fTPPlayProb[action] * fInitialTPProb[i];
  ta = fTAPlayProb[action] * fInitialTAProb[i];
  lp = fLPPlayProb[action] * fInitialLPProb[i];
  la = fLAPlayProb[action] * fInitialLAProb[i];
  // Sum all the separate probabilities, to create the
  // normalizing constant, Pr(B), and normalize each
  // value so that the probabilities sum to 1
  float prob = tp+ta+lp+la;
  if (prob < 1)
  {
    tp = tp / prob;
    ta = ta / prob;
    lp = lp / prob;
    la = la / prob;
  }
  // Set all initial probabilities as the new value,
  // Pr(A|B)
  fInitialTPProb[i] = tp;
  fInitialTAProb[i] = ta;
  fInitialLPProb[i] = lp;
  fInitialLAProb[i] = la;
}
```

Fig. 2. Pseudocode of the Bayes' Theorem predictor

The calculation uses an initial probability of 0.25 as Pr(A) for the first iteration, as each opponent style is assumed equally likely when no actions have been analysed.

The values used to represent the probability of a player of style A taking an action B were evaluated by analysing the past actions of players of that style over thousands of games; the probability that a loose aggressive player would make a betting action, for example, was calculated by recording the frequency of betting actions made per 100,000 games. These probabilities can be seen in Table III. Actions that could not be utilised in defining opponent type were ignored, such as most actions taken when not under any monetary stress.

TABLE III
ACTION PROBABILITIES FOR EACH OPPONENT STYLE

	FOLD	CHECK /CALL	BET /RAISE
LA	0.36	0.05	0.59
LP	0.60	0.29	0.11
TA	0.73	0.02	0.25
TP	0.87	0.07	0.06

The updated initial probability represents the probability that the player is of each of those styles. Fig. 3 gives an example that shows how quickly the analysis player probabilities converge.

If these new probabilities do not exceed a minimum desired probability for an opponent to be defined as a specific type (in these experiments, a value of 0.95 was used), the probabilities are then re-used on the next cycle. This raises an issue: if the probability of a player of type A taking action B is set high, the "Learned" probability of being type A could breach the threshold of certainty very quickly, and possibly lead to an erroneous analysis. Furthermore, setting the probabilities too low will leave a lot of uncertainty about defining a player's type, reducing the convergence speed.

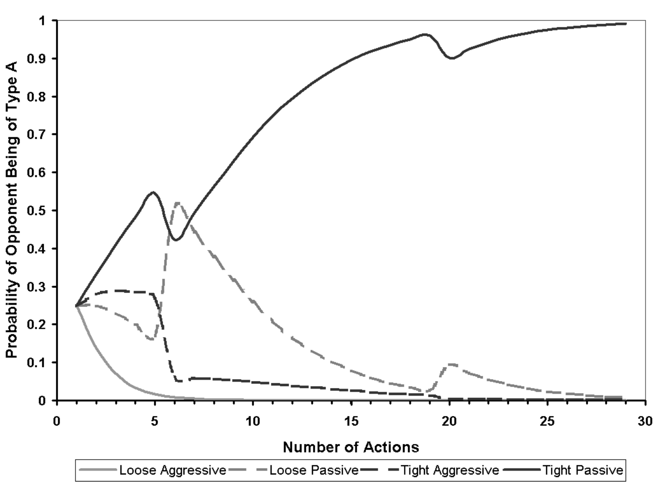

Fig. 3. Graph displaying how player type probabilities alter given opponent actions. In this case, the analysis player converges to the conclusion that the opponent is a tight passive player.

D. Modeling Correct Responses for Opponents of Differing Styles

The four Anti-Players combine to form the strategies of the Analysis Anti-Player. After the analysis player learns the opponents' styles, it prioritizes its reactions in relation to the most "dangerous" opponent type. For example, a tight passive player's actions would be taken more seriously than that of a loose aggressive player. This risk analysis leads to choosing a specific anti-player's tactics dependent on the greatest threat. A tight passive player has the greatest priority, due to the tight risk-free nature of play. After this, a tight aggressive player would be given priority. When considering only loose style players, the greatest popularity of a style is given precedence, pigeonholing the entire set of opponents in relation to that specific style. Fig. 4 shows how the Analysis Anti-Player chooses its tactic. The player analyses each of the opponents' actions and determines the opponents' play style using the algorithm defined in Fig. 2. It then analyses the assumed styles of all the opponents, and chooses to play against the most threatening style; tight passive and aggressive players take priority as tight players are much stricter in their style of play, if no tight players exist, then priority is given to the loose style used by most players. When the Analysis Player has decided which style of player to respond to, it will use the relevant 'Anti' style of play.

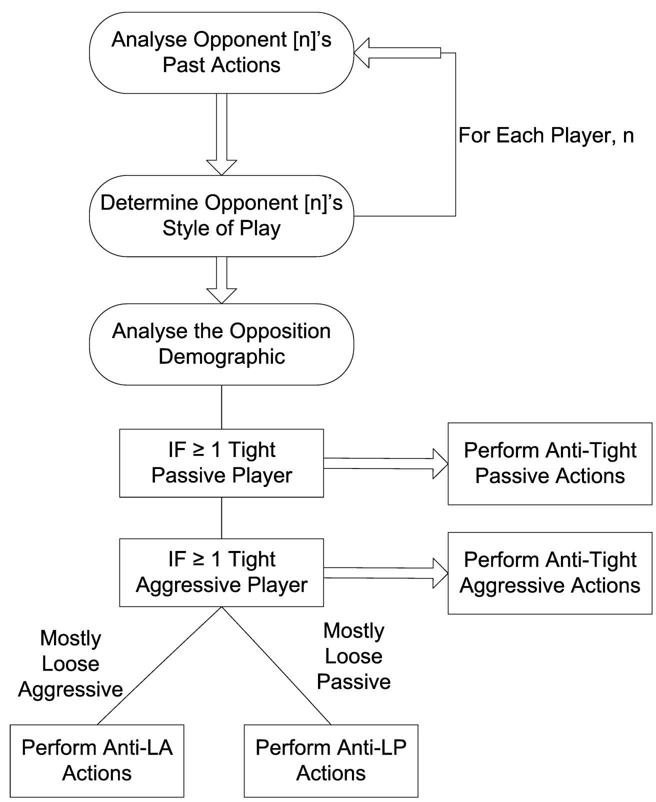

Fig. 4. Diagram representing the flow of actions with an Analysis Anti-Player to determine the opponents' style and choose which tactic to employ

E. The Simulation Player

The last player to be created was what is called the Simulation Player. This player, on each of it's turns is told the style of each opponent, and it then runs a simulation of the game within itself (loading a new instance of the game class) with the current demographic of players, and runs tests against the players over a discrete set of 66 α and β values in increments of 0.1 where $0 \leq \alpha \leq \beta \leq 1$. These tests consist of 100 games per [α, β] pair. It then utilizes what it sees as the 'best' values of [α, β] to make subsequent decisions against the opponents. The graphs in Figures 6 – 9 show a representation of the results that a simulation player receives when playing different sets of opponents.

IV. Results And Analysis

The comparison of the Analysis Anti-Player against the Simulation Player [Fig 5] represents an average of the percentage of tournaments won by each player against every combination of opponents in a four-player environment. All experiments are run on a Pentium IV 3.0 GHz HT with 1GB RAM using C#.NET running under Windows XP SP2.

We observed that the averaged results were not sensitive to the order of opponents around the table, and win percentages had only a standard deviation of 3%. We can see that the performance of the Analysis Anti-Player is comparable to, and on occasion surpasses that of a player that is already knowledgeable of the opponent's styles. Due to the random nature of the hands, it is not surprising that the Analysis Anti-Player is occasionally the best. These results show how successful the pigeonholing technique is compared to a 'custom built' design that the Simulation Player creates for the current demographic. There is a point where the success rate of the Simulation Player reaches as low as 40%. It should be appreciated however, that 40% is not necessarily a failure as this is a four-player game, and 40% is greater than a 'fair share' of tournament wins.

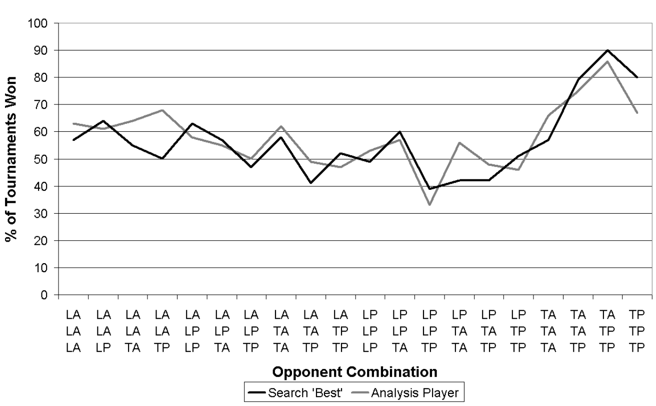

Fig. 5. Graph displaying combinations of opponent styles against the percentage of tournaments won by each player

One comparison between the two players is related to the performance of each player against the groups of mostly tight players on the right hand side of the graph. These results show that the players are both quite competent against these styles of play, and the Analysis Player has nearly an 85% success rate. An explanation for the high rate of success, even against four players is possibly due to the tight nature of the opponents; any action that the player takes which is not a folding action may steal the pot from a tight player with a better hand: tight players are very susceptible to bluffing.

The success against mostly loose players is slightly less prominent, but still very impressive, averaging around 55% of tournament wins. This is not surprising since a loose player's actions reveal relatively little about the card held. This can lead to situations where a loose opponent will have a very strong hand, and the Analysis Players will still remain in the hand, resulting in an unsuccessful tournament.

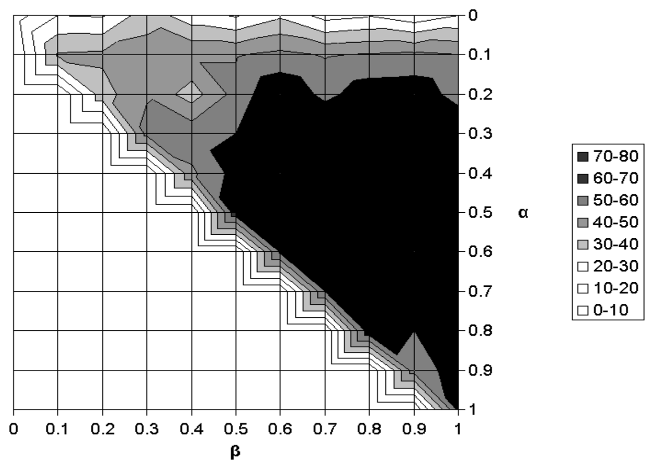

Fig. 6. Graph displaying the tournament success of α against β (in % of tournaments won) when playing a four-player tournament against LA/LA/LA opponents

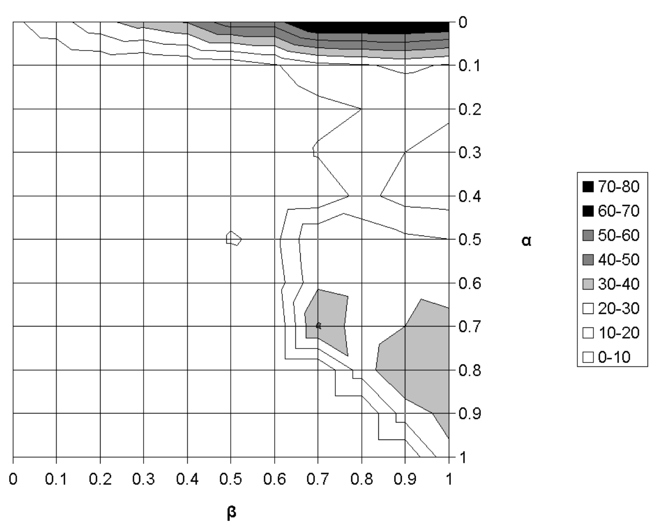

Fig. 7. Graph displaying the tournament success of α against β when playing a four-player tournament against TP/TP/TP opponents

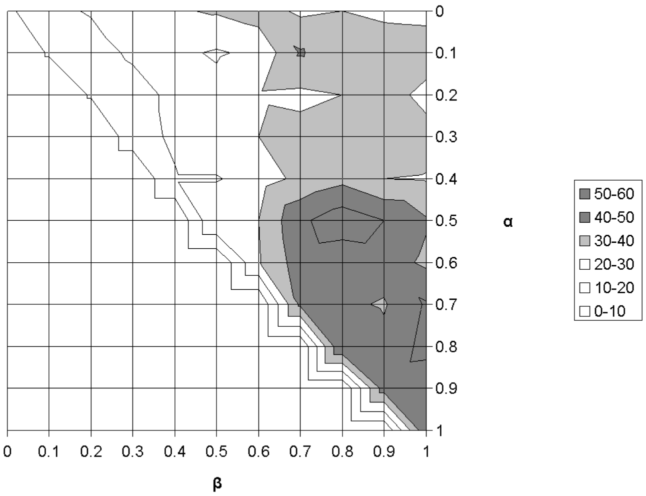

Fig. 8. Graph displaying the tournament success of α against β when playing a four-player tournament against LA/LA/TP opponents

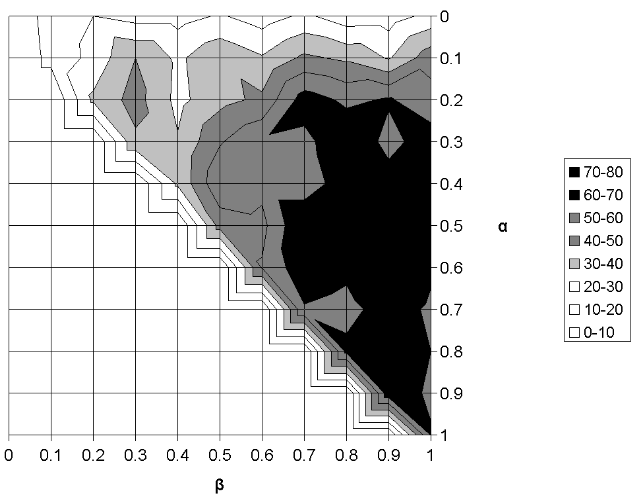

Fig. 9. Graph displaying the tournament success of α against β when playing a four-player tournament against LA/LA/LP opponents

Figures 7, 8, and 9 provide some explanation as to why the pigeonholing technique works; these figures show the success of each possible value of α and β against a different collection of opponents; the darker areas indicate a greater concentration of large win percentages.

Fig. 6 shows the successful probabilities against three loose aggressive opponents, a large area of high success values can be seen, indicating that playing tightly, as well as playing with a low amount of belief in the severity of the opponent's actions can bring a large number of wins. Fig 7 shows the successful probabilities against three TP opponents, and the graph shows that the concentration of wins is very low, mainly due to the low probability of a large pot when playing a tight player, especially when playing against a tight passive player. The graph indicates that the best strategy is to participate in as many hands as possible, but not to make a betting action unless completely certain of a win. This appears to be a rational tactic, as remaining in a hand increases the probability of stealing a pot from a tight player.

Fig. 8 shows the effect of replacing one loose aggressive opponent with a tight passive one. The transformation from Fig. 6 to Fig. 8 is quite dramatic, and exemplifies the effect that a tight passive player can have upon a game of three loose opponents, in turn justifying the pigeonholing technique, as the resulting graph is much more closely related to the graph of three tight passive opponents than it is for three loose aggressive opponents. The main explanation of this is probably due to the order that a game may take with this general demographic; the loose players would have a tendency to risk many chips, and may get removed from the tournament early, leaving only the Anti-Player and the tight passive opponent, which would probably dominate most of the games. It should also be noted that the number of tournament wins in total across the graph is relatively low compared to other graphs. This is however, one of the points where the Analysis Player greatly outperforms the Simulation Player (by nearly a 20% margin). This is probably due to the Analysis Player modifying its behaviour while playing to deal with the greatest threat. When the loose players have been removed, the main focus of play will be against the tight passive player.

Fig. 9 shows how small an effect is brought upon the same demographic of loose aggressive players by adding a loose passive player. As can be seen, there is very little difference between the graph in Fig. 9 and that of Fig. 6, which is mainly caused by the loose nature of the opponents, as the dark area of the graph (displaying an 'area of believability') shows how fragile a loose player is against partially tight play.

V. Conclusion

A player is presented for a simplified game of poker, which analyses the past actions of its opponents, and in turn uses Bayesian probability ideas to learn what style of opponents it is facing.

The player uses a system of ranking to determine the greatest threat and acts as if all opponents are of that style by 'pigeonholing'. The performance of the learning player compared to one that knows the opponent styles without the need for probabilistic calculation is quite similar, and demonstrates the effectiveness of both learning and pigeonholing.

A single deterministic opponent proves simple to beat when using the Bayesian predictor function. The quick convergence shown in Fig. 3 illustrates the difficulty that this current predictor would face when playing against a dynamic opponent.

Future work will investigate the situation where the predictor function is able to cope with partially randomised, dynamic, or bluffing players. These future players would use stronger play policies than the current insensitive, and non-reactive opponents. In these cases, an opponent could, accidentally or deliberately, hide its playing style, and make the Bayesian approach to learning more difficult.

References

[1] Parlett, D. *Oxford guide to card games,* USA: Oxford University Press, 1990

[2] Reuters. U.S. ban sparks web gaming crisis [Online] available from<http://www.cnn.com/2006/BUSINESS/10/02/uk.betting.reut/index.html>, 2006

[3] Campbell, M. Haone Jr., A.J. Hsu, F-h. Deep Blue, *Artificial Intelligence,* 2002, Vol.134, (pp.57-83)

[4] Koller, D. Pfeffer, A. Generating and solving imperfect information games *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI),* 1995, (pp. 1185-1192).

[5] Sakaguchi, M. Sakai, S. Solutions of some three-person stud and draw poker. *Mathematics Japonica* 1992, (pp. 1147-1160)

[6] Billings, D. Burch, N. Davidson, A. Holte, R. Schaeffer, J. Schauenberg T, and Szafron, D. Approximating game-theoretic optimal strategies for full-scale poker *In Proceedings of the eighteenth International Joint Conference on Artificial Intelligence* 2003, (pp. 661-668).

[7] Billings, D.; Davidson, A.; Schaeffer, J.; and Szafron, D. The challenge of poker. *Artificial Intelligence Journal,* 2002, (pp. 201–240).

[8] Schaeffer, J. Billings, D. Peña, L. Szafron, D. Learning to play strong poker *In ICML-99, Proceedings of the 16th International Conference on Machine Learning,* 1999

[9] Billings, D. Papp, D. Schaeffer, J. Szafron, D Opponent modeling in poker *Proceedings of the fifteenth national/tenth conference on Artificial intelligence/Innovative applications of artificial intelligence,* 1998, (pp. 493 - 499)

[10] Saund, E. Capturing the information conveyed by opponents' betting behaviour in poker. *In proceedings of 2006 IEEE Symposium on Computational Intelligence and Games (CIG),* (pp. 126-133)

[11] Koller, D. Pfeffer, A Representations and solutions for game theoretic problems *Artificial Intelligence*, 1997, (pp.167–215)

[12] Burns, K. Style in poker, *In Proceedings of 2006 IEEE Symposium on Computational Intelligence and Games (CIG),* (pp.257-264)

[13] Barone, L. While, L. An adaptive learning model for simplified poker using evolutionary algorithms *In proceedings of Congress of Evolutionary Computation 1999* (CEC'99), July 6-9, Washington DC, (pp 153-160).

[14] Southey, F. Bowling, M.P. Larson, B. Piccione, C. Burch, N. Billings, D. Rayner, Bayes' Bluff: Opponent Modelling in Poker. *In Proceedings of the 21st Annual Conference on Uncertainty in Artificial Intelligence* (UAI-05), 2005, (pp 550-555)