Player Profiling in Texas Holdem

Karl S. Brandt
CMPS–240, Spring 2004
kbrandt@cs.ucsc.edu

1 Introduction

Poker is a challenging game to play by computer. Unlike many games that have traditionally caught the interest of the Artificial Intelligence community, poker is a game of uncertain information. Where in games like chess and checker have the entire game state is known to all players, in poker only the part of the game state is known. Since searching the game space is insufficient to make good playing decisions, a poker player must choose based on probabilistic knowledge. This both makes poker a difficult problem to solve and an interesting avenue for artificial intelligence research.

Average human players might very well have a tough time playing against a purely probabilistic player simply because odds are difficult to calculate on the fly. Nevertheless, playing based purely on probabilities is not enough to defeat expert human players. These level of players have good intuitive feeling for what the odds are at any given point in the game. Once they recognize an opponent's style of play, they are able to adapt their tactics based on the likely actions of their opponents. These expert players essentially begin to profile their opponent's behavior and act so as to subvert their opponent's strategies.

Bluffing is the most obvious example of this. Instead of folding poor quality hands, a bluffer would bet big at random intervals. This would mislead a probabilistic player into believing the bluffer has high value cards and as a result fold otherwise winning hands.

2 Related Work

The Computer Poker Research Group at the University of Alberta has been the most active in researching the playing of Texas Holdem poker. Their early work produced *Lokibot* [2], a basic probabilistic player. They calculated HandStrength (HS), which was the probability that a given hand would beat a single opposing hand. They also added calculations for hand potential which took into account the possibility of improving a given hand. Combined, these were termed Effective Hand Strength (EHS). EHS was used to determine the players actions based on the expected value of a given hand when looking at the cost of calling or raising and the size of the pot.

Billings *et al* [1] later added opponent modeling to *Lokibot*. This involved both weighting the probabilities for all opposing hands and tracking opponent behavior. At a given game state, hands that were very likely to be played up to that point were given high weights, where opponents were likely to have folded were given very low weight. Also, probability triples of {fold,call,raise} were tracked for repeated situations in the game. So if a player raised in a situation where they would have folded with an average hand, it could be inferred that had an above average hand. Then all below average hands would have their weights reduced to near zero so they would not enter into the decision making.

Still later, Billings *et al* [3] implemented a neural network to explore various aspects of opponent modeling. Due primarily to its computational demands, it was not intended to be an online means of profiling. Instead, it was used to reveal what aspects are most important in opponent modeling. For instance they discovered that an opponent's previous action and amount to call were important aspects to model that they had not previously included.

3 Previous Work

In our first project [4], we wrote a poker playing game. We also developed several bots for playing poker.

The *ProbOnly* player makes decisions purely probabilistically based on hand strength. Since hand strength is hard to accurately gage in the Blind round, all our tested bots use the same heuristic for making Blind round decisions. This heuristic is largely based on the one used in the starter bot for playing on the University of Alberta's online matches.

We also wrote two bluffing players that were both able to defeat *ProbOnly*. Since it was the better of the two, *Nbluff* was used as an opponent to test newer profiling bots in this paper.

4 Profiling

4.1 Eliminating Unlikely Hands

Since simple probabilistic play, will overestimate the probability of winning due to having included hands that are likely to be folded early, we decided to try and reduce the importance placed on poor hands. We built a 52X52 table of all possible hands and tracked the frequency with which each hand was played. The easiest thing to do was to watch the behavior of our own player. Essentially, we wanted to know the probability of a given hand being played past the flop round and assign a weight for that hand. For instance if 7,2 unsuited was always folded, we would calculate the weight to be 0 for that hand and disregard it when calculating hand strength. On the other hand, a pair of Aces was always played so its weight would be 1.

Figure 1 shows a partial weight table. For brevity, we are only showing hands that where both cards are hearts. Most hands get values of 1.0. Those hands that are frequently folded get values less than 1.0. For clarity, Figure 2 shows the same data except that it is displaying $(1 - \mathit{weight})$. This shows that some hands are not always played past the Blind round and have lower weights.

Unfortunately, this was the first of many failures. When two players were played against each other with one using this weighting scheme and the other not, the one without using weights consistently won. It is apparent that the calculations of HS were somehow getting skewed incorrectly from this. It is possible that these weights should be normalized across the whole table. So instead of the table entries representing weights, they would represent the probability of that particular

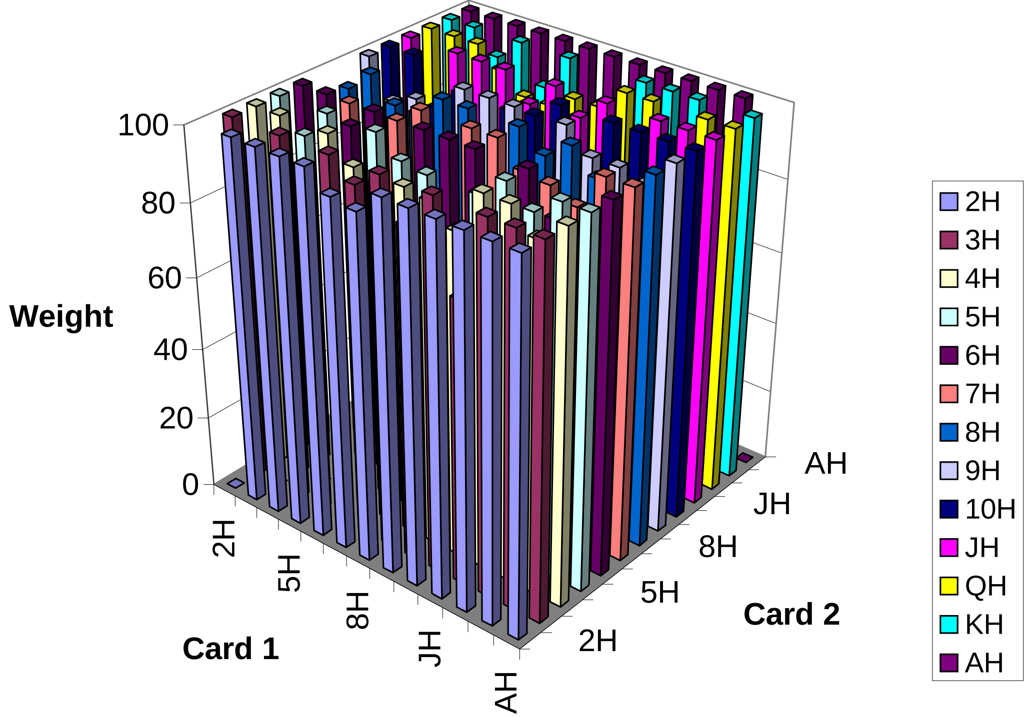

Figure 1: Example of weights for suit of hearts.

hand being played. HS would then be calculated by summing up the probabilities for all hands that a player's hand would beat.

4.2 Equivalence Groups

Next we attempted to alter these weight tables based on a player's frequency of staying in a hand after the Blind. This was done with the thought that folding after the Blind is relatively infrequent when compared to the number of folds before the Flop.

If a player folds 60% of the Blinds, one could infer that the worst 60% of the possible hands are never played. With 1326 possible two card hands, it would have been nearly impossible to rank all hands by their value. Even when looking at equivalent hand types there are still 169 classes of hands that would need to be ranked. Instead, we chose to use the grouping system developed by Sklansky [5], a noted author on Texas Holdem poker. Sklansky classified hands into eight different Groups. Group 1 hands being the very best and Group 8 hands being the ones that should only rarely be played. There were a considerable number of unclassified hands that we assigned to an additional Group 9. Essentially, these hands should never be played by a rational player.

In Figure 3, the X-tics mark the divisions between groups. Group 1 is the best 2% of hands;

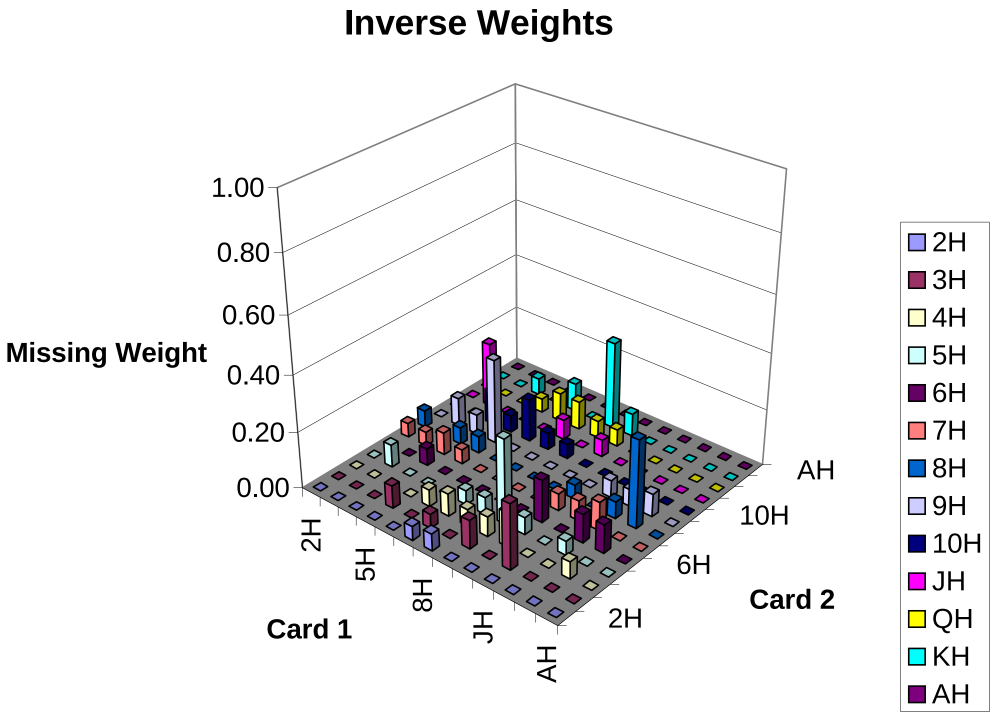

Figure 2: Inverse of weights for suit of hearts.

Group 9 is the worst 60% of hands. If a player is observed to call some percentage of the Blinds, all hands below that group are assumed to be played and all hands above that group are assumed to be folded. The group where the percentage falls are given a prorated weight. Figure 3 shows examples for 35% and 50% call rates.

5 Results

Unfortunately, the results were universally disappointing. The profiling player was never able to beat either *ProbOnly* or our *Nbluff* player. Figures 4, 5, 6 and 7 track players bankrolls over time. In these games, the ”small” bets were 10 and the ”large” bets where 20.

5.1 Probabilistic vs. Bluffer

Since our previous project was looking at no-limit poker, we wanted to confirm that the *Nbluff* was still able to defeat *ProbOnly*. Figure 4 shows the results of starting *ProbOnly* and *Nbluff* both with bankrolls of $10,000. While *ProbOnly* briefly led in the beginning, *Nbluff* steadily won over all with an average of 0.027 small bets per hand.

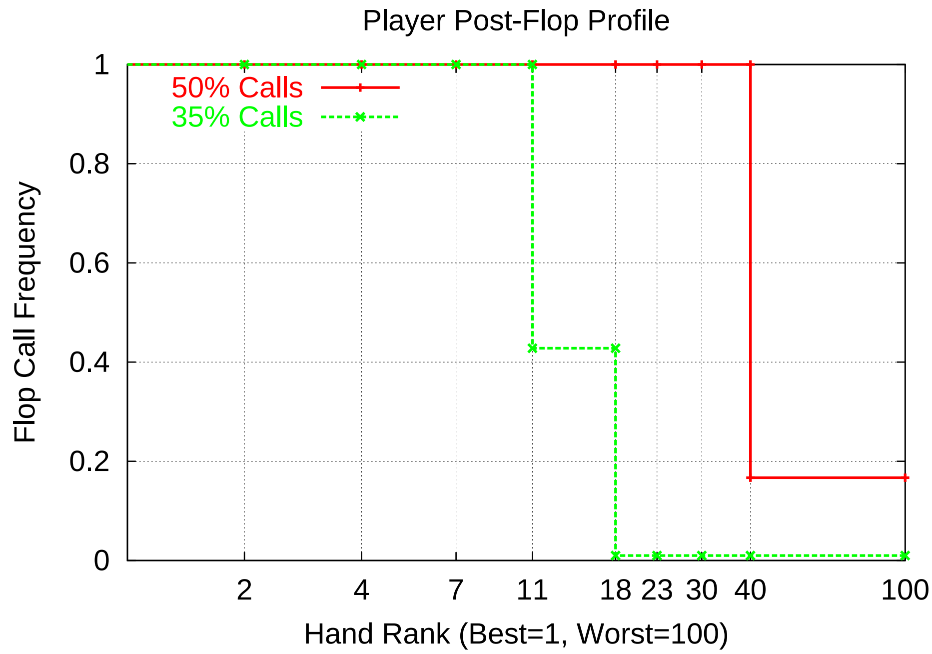

Figure 3:

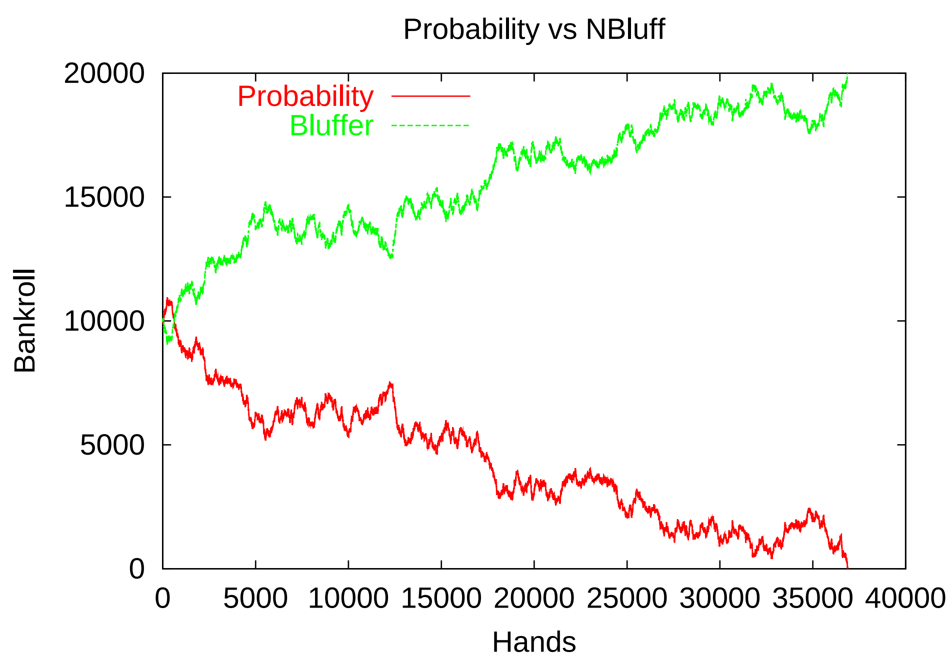

Figure 4:

5.2 Profiler vs. Probabilistic

Figure 5 shows the results of our profiler playing against *ProbOnly*. Both players were given initial bankrolls of $5000. While the profiler was able to get back to even at after around 4000 hands, it

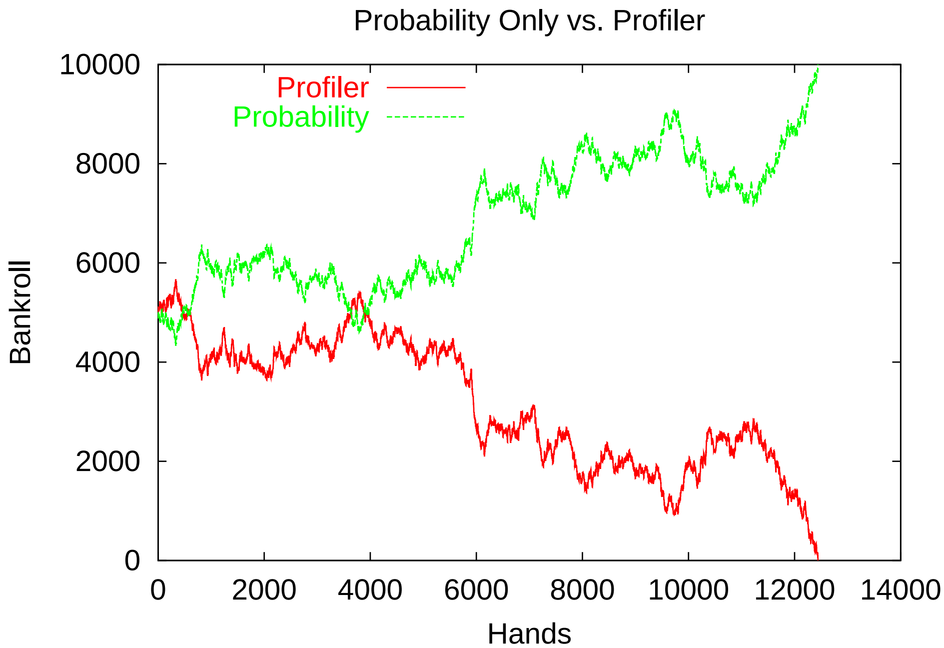

Figure 5:

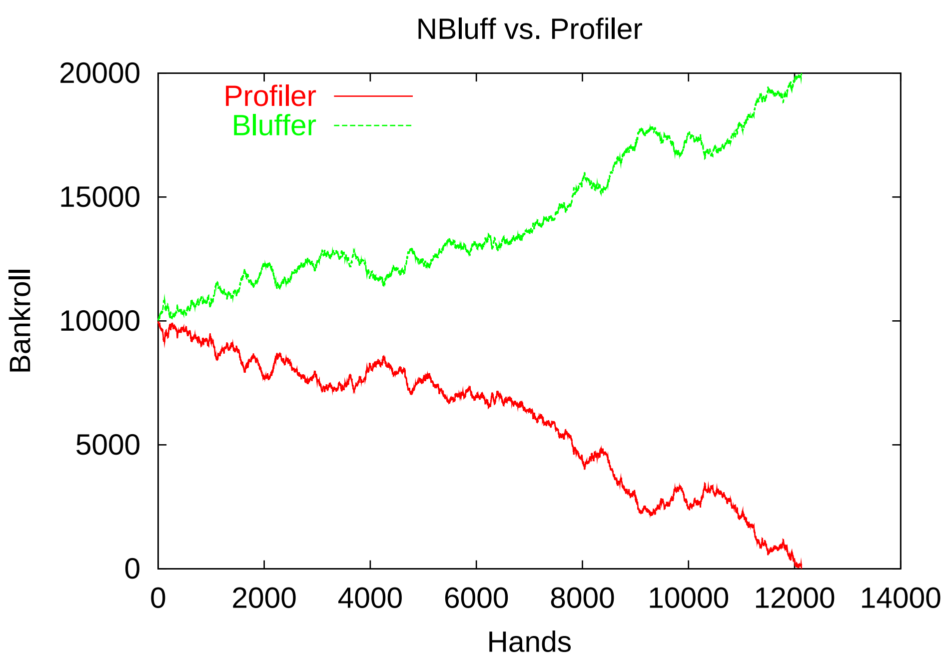

Figure 6:

lost in the long run. By the time the profiler was broke it lost an average of 0.160 small bets per hand.

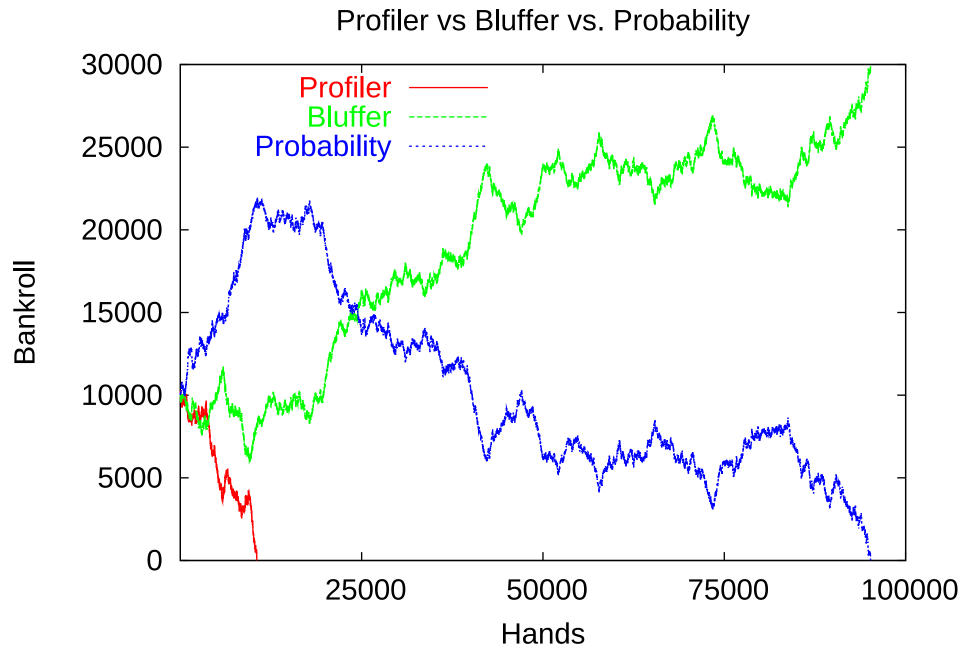

Figure 7:

5.3 Profiler vs. Bluffer

The original hope in this project was that a profiling player would be able to defeat a bluffer. The profiler would detect that the bluffer was playing more hands than would normally be played and infer that low ranked hands were being bet up.

Figure 6 shows the result of a match between our profiling player and *Nbluf* with both players bankrolls starting at $10,000. Once again, *Nbluf* steadily defeats the profiler. The profiler did somewhat better than against *ProbOnly*, only losing an average of 0.082 small bets per hand.

5.4 Profiler vs. Probabilistic vs. Bluffer

We were also interested in testing if there were any interactions between the players when all of them were at the table. Figure 7 shows the results of all three players starting with a bankroll of $10,000. Of course, the profiling player lost first at an average rate of about 0.095 small bets per hand. Interestingly, *ProbOnly* was the primary beneficiary of the profiler's poor play. It took *Nbluff* over 95,000 hands to finally defeat *ProbOnly*.

6 Conclusions

Both *ProbOnly* and *Nbluf* are good enough players that it is difficult to come up with schemes to beat them. The profiler we built to that ends is clearly flawed and was unable to defeat either previous player. The profiler did slightly better against *Nbluf* than *ProbOnly* despite *Nbluf* being a better player.

Clearly, something has to be done to fix the profiling player. Our current guess is that hand strength is no longer being calculated correctly. This would seriously handicap any player, even one that was benefiting from profiling. At least, this is the hoped explanation as to why the results were so disappointing.

References

[1] Darse Billings, Denis Papp, Jonathan Schaeffer, and Duane Szafron. Opponent modeling in poker. In *Proceedings of AAAI-98 (15th National AAAI Conference)*, 1998.

[2] Darse Billings, Denis Papp, Jonathan Schaeffer, and Duane Szafron. Poker as a testbed for machine intelligence research. In *Artificial Intelligence'98 (Canadian Society for Computational Studies in Intelligence)*, 1998.

[3] Darse Billings, Denis Papp, Jonathan Schaeffer, and Duane Szafron. Improved opponent modeling in poker. In *Proceedings of the 2000 International Conference on Artificial Intelligence*, pages 1467–1473, 2000.

[4] Timothy Bisson, Karl S. Brandt, and Nate Edel. Simulating and playing poker.

[5] David Sklansky and Mason Malmuth. Two Plus Two Publishing, 2nd edition, 1994.