

Pokerbots strategy report - Team q

David Lu

January 28, 2015

Team *q* is named after a team of the same name that competed in a different coding competition (Battlecode) two years ago. David Lu and Alex Chen were the two members of that team; Alex Chen is a member of the current Team q as per tradition, but did not actually participate in Pokerbots.

The primary player of Team q is named qbot. There are two key elements that work together to allow qbot to function - a hand strength calculator and a player profiling system. More minor elements are the bluff evaluation, out detection, and byte packing systems.

1 Hand Strength Calculator

The hand strength calculator is used to determine the strength of the cards that qbot currently holds on to. I created a tool that, for each of the 169 distinct preflop hands, ran 100000 random showdown simulations against two other random preflop hands and recorded the number of showdowns the preflop hand won. These numbers were used to rank the preflop hands - where the 'strength' of a preflop hand was its probability of winning.

A more dynamic technique was necessary for situations postflop where there were cards on the board, as I didn't particularly want to run simulations for every possible combination of hand \times up to five board cards. Therefore, the 'strength' of a postflop hand was calculated to be the relative strength of that 2-2-card hand given the board compared to the strength of all other possible 2-card hands given the board. The best possible 2-card hand given a certain board would have a relative strength of 1 on the board, and the worst possible 2-card hand would have a strength of 0. An additional benefit of this system was that the relative strength of a hand was the same as the probability that hand would win an 'instant showdown' against another randomly-selected hand.

2 Player Profiling System

The player profiling system allows qbot to play effectively against many kinds of strategies and keeps it from being exploited. qbot creates a profile for each player it encounters. Any time that player makes an action, that action is recorded in the profile. This allows qbot to keep track of statistics about the opponent, for instance how often the opponent bets when it is checked to him during the flop, or how often the opponent folds to a bet preflop. In addition, for each bet or raise made, the system stores the relative size of the bet or raise (as compared to the maximum bet/raise that was possible for that action). This allows q to estimate the strength of an opponent's hand based on the opponent's past statistics. For example, an opponent who bets rarely will be evaluated to have a very strong hand when he does bet, whereas an opponent who bets whenever possible will fold preflop almost all of the time will be assumed to have a pretty good hand when he does not

do so. Storing the relative sizes of bets and raises allows further analysis of opponent actions - for example, an opponent who min-bets half the time and max-bets the other half will be judged to have a stronger hand when he max-bets versus when he min-bets.

Each time qbot is able to make an action, it looks at the past actions of the opponents and uses the opponents' profiles to estimate the strength of the opposing hands. Then, it responds based on the strength of its own hand in comparison to the strength estimations of its opponents' hands. This is the core of qbot's decision-making; it also involves certain numeric thresholds (for example, raise if your hand strength is greater than the average of 0.9 and the opponent's estimated hand strength). These numeric thresholds were initially guesses, but were later improved by the use of genetic algorithms to find the best values for the thresholds.

3 Bluff Evaluation

qbot decides whether to bluff based on its position and the statistics of the opposing players. If the opposing players fold enough so that a bluff will result in positive expected value, qbot will bluff. In addition, qbot occasionally attempts a steal if it has position and all other players have checked. It records the result of these steals, and becomes less likely to steal as the failure rate of the steals declines.

Finally, if an opponent player bets much more than it calls, qbot will occasionally attempt a bluff reraise. Again, the frequency of this bluff depends on the statistics of the opponents.

4 Out Detection

Any time qbot would have folded a hand, it checks its outs - defined by how likely it would be that the next card to be dealt would put its relative strength significantly higher than its estimation of the opponents' strengths. If the likelihood of obtaining an out is higher than the amount q would need to add to stay in the hand, q will change its decision from fold to call.

5 Byte Packing

qbot uses the keyvalue storage system to store its player profiles and bluff success records between matches. However, a naive implementation of this resulted in strings of digits and spaces that were so long that the total 2000 bytes allotted would be exhausted if profile data were stored for each team. Therefore, q compresses this data before putting it in the keyvalue storage. All the data is composed of the ten digits along with spaces. These are broken into chunks, which are converted into longs by considering the chunks as base-12 numbers (where spaces are the '10' digit and zeroes are the '11' digit). These longs are converted into base-94 numbers using 94 different ASCII characters, and these characters are stored as key values. When retrieved, the process is reversed to turn the ASCII string into a string of digits and spaces. This process allows the same amount of data to be stored using less characters, and can store n digits of numbers in roughly $\frac{n}{2}$ characters.