6.176 Strategy Report

Str8Rippin

Erwin Hilton, Richard Spence, Gabriel Esayas

January 30, 2015

Introduction

This strategy report outlines the strategies that our pokerbot used for the 6.176 Pokerbots competition, including how we classified other opponents to determine whether to raise, check, call, or fold.

The `Opponent` class

We created a Python class `Opponent` that keeps track of player stats throughout the game. Thus, whenever `NEWGAME` is received, we create three `Opponent` objects, including one for our own player. Each opponent has several basic attributes, including name, stack size, what seat the opponent is at (1, 2, or 3), whether the opponent is playing the current hand, and whether the opponent is eliminated from the game. Additionally, each `Opponent` has the following attributes:

`bot.foldPer`	percentage of hands folded pre-flop
`bot.folds`	number of folds pre-flop
`bot.VPIP`	percentage of hands in which opp. put money in pot pre-flop
`bot.PFR`	percentage of hands in which opp. raised pre-flop
`bot.WTSD`	percentage of hands in which opp. went to showdown
`bot.WMSD`	percentage of showdowns in which opp. goes and wins
`bot.AFflop`	aggression factor - flop, defined as (# bets+ # raises)/(# calls)

These attributes allow us to classify an opponent one of several categories: nit, shark, bomb, maniac, calling station, cashcow, loose, and unknown.

We classified players as follows:

$$
\texttt{bot.playerType} = \begin{cases}
\text{NIT} & bot.VPIP \leq .13 \text{ and } bot.PFR \leq .13 \\[1em]
\text{SHARK} & .1 < bot.VPIP < .3 \text{ and } 0.05 \leq bot.PFR \leq .25 \text{ and} \\
& bot.AFflop \geq 2.5 \text{ and } bot.WMSD \geq .4 \\[1em]
\text{BOMB} & self.VPIP \geq .4 \text{ and } self.PFR \geq .13 \text{ and } self.AFflop \geq 5 \\[1em]
\text{MANIAC} & self.VPIP \geq .35 \text{ and } self.PFR \geq .25 \text{ and } bot.AFq \geq .5 \\[1em]
\text{CALLING STATION} & .2 \leq bot.VPIP \leq .4 \text{ and } bot.PFR \leq .15 \text{ and} \\
& bot.WTSD \geq .35 \text{ and } bot.AFflop \leq 3 \\[1em]
\text{CASHCOW} & bot.VPIP \geq .35 \text{ and } bot.PFR \geq .25 \text{ and } bot.WTSD \leq .38 \\
& \text{ and } bot.AFq \leq .5 \\[1em]
\text{LOOSE} & bot.AFq \geq .4 \text{ and } bot.PFT \geq .13 \\[1em]
\text{UNKNOWN} & \text{otherwise}
\end{cases}
$$

The following table gives a brief description of each player type:

NIT	based off VPIP and PFR, nits voluntarily bet a low percentage preflop; usually only play premium hands
SHARK	hard to exploit; choose hands well and are aggressive about them postflop; have a bit wider range of playing hands than a nit, but still tight
BOMB	loves to play hands aggressively; plays very loosely
MANIAC	similar, very, very loose player, usually know they will play their hand
CALLING STATION	love to call, but usually do not bet or raise; usually never bluff (only bet/raise on very good hands)
CASHCOW	play loosely but not aggressive postflop; can usually make them fold postflop
LOOSE	general term for a player who plays a large amount of hands but was not classified any of the above types

We decided to only take player types into account after 10 hands have elapsed, since it is difficult to make choices based on player type early on.

We also use the M-ratio, so a player's M-ratio (denoted `bot.MRatio`) is defined by

$$
M = \frac{\text{stack}}{\text{small blind} + \text{big blind} + \text{total antes}}
$$

However, the M-ratio simplifies in this game, since the antes are zero, and the small blind is always 1 and the big blind is always 2. Hence, the M-ratio is simply $\frac{\text{stack}}{3}$. Even though M-ratio is not very interesting in this tournament, it gives some indication of how many hands we can survive.

The equity calculator

We used the given equity calculator, which takes in an input our hole cards, any community cards, as well as possibly the opponents' hole cards, and outputs a number in $[0, 1]$ indicating the expected amount of the pot we should win with our hand.

Afterwards, we computed the estimated value (EV) for each hand, given by

$$EV = \text{equity} \cdot \text{winAmt} - (1 - \text{equity}) \cdot \text{loseAmt}$$

where winAmt equals the sum of the opponents' contributions to the pot, and loseAmt is the amount we have contributed to the pot.

Fold expected value is equal to

$$\text{Fold EV} = \text{opp's fold \%} \cdot \text{potSize} + (1 - \text{opp's fold \%}) \cdot EV$$

This is essentially our expected value, if the opponent folds in the current round. We solved for the maximum bet amount so that Fold EV ¿ 0, and used this as our maximum bluff amount. We usually bluffed when our opponents were classified NIT, and they were not raising, or if they were classified CASHCOW and were not raising after the flop, since we believed we can make them fold postflop.

Strategy

At the end of each hand, we update the stats that were described above, for each player, if that person played that hand.

First, we parse the inputs from each `GETACTION` packet. Then we use the given equity calculator to determine the hand equity.

Preflop

We mostly based our decision on our equity during preflop. If our equity is low, then we faced a few options:

- If a nit is raising, then he must have an excellent hand, so we check/fold.

- Otherwise, with probability 0.35 and if the minimum bet amount is small, we bluff.

- Otherwise, we check/fold.

Otherwise, if $EV > 0$, we look at equity. If our equity is larger than a threshold, we bet or raise. If $EV < 0$, we check/fold.

Postflop

Generally, we will always bet, raise, check, or call when $EV \geq 0$. We exploited cashcows if they were the only players left in the hand by bluffing.

If $EV < 0$ and earlier this round, a shark, nit, or calling station bets or raises, we will check/fold, since the bet/raise likely indicates he has a very strong hand.

All-in

We made an exception on our strategy when our M-ratio becomes very low. We decided to go all-in if our M-ratio is less than 4, or if our M-ratio is between 4 and 7 and our equity is at least 0.6. If our M-ratio is less than 7 and we have poor hand equity, we simply fold.

Reflections

Our bot played well with certain hands, but further improvements to our bot in the future could include: better classification so that all player types are classified appropriately, and storing player information from previous matches so that we can use the information in future matches to classify them better.