Subject:

Computer games  dr inż. Bartosz Ziółko

Authors:

Grzegorz Rajczyk  rgrzegor@student.agh.edu.pl
Maciej Kurek  kumaciej@student.agh.edu.pl
1. Introduction

The task was to write a computer bot to very popular card game – Poker Texas Hold’em. Texas Hold’em is a variation of the standard card game of poker. The game consists of two cards being dealt face down to each player and then five community cards being placed face-up by the dealer—a series of three ("the flop") then two additional single cards ("the turn" and "the river"), with players having the option to check, bet, raise or fold after each deal. Hold'em is normally played using small and big blind bets – forced bets by two players. Antes (forced contributions by all players) may be used in addition to blinds, particularly in later stages of tournament play. A dealer button is used to represent the player in the dealer position; the dealer button rotates clockwise after each hand, changing the position of the dealer and blinds. The small blind is posted by the player to the left of the dealer and is usually equal to half of the big blind. The big blind, posted by the player to the left of the small blind, is equal to the minimum bet.

2. Issues

One of the most important things in Texas Hold’em is knowing how to evaluate a hand. The strategy of playing each hand can be very different according to the strength of the hand, for example on a strong hand a player might want to try to appear weak in order not to scare off other players with weaker hands and on a weak hand a player might try to bluff other players into folding. In our algorithm only thing we have to know is which of two hands wins, so we used a hand comparator class from the Poker Game Engine.

3. Algorithm and implementation

3.1. Preflop

“Preflop” is a situation where two cards were dealt to each player, and there are no cards on the table. In our strategy we use a “Sklansky and Malmuth hand group” to decide how to play at Preflop. David Sklansky and Mason Malmuth assigned each possible hand to a group, and proposed all hands in the group could normally be played similarly. Stronger starting hands are identified by a lower number. Hands without a number are the weakest starting hands. The better hand we have the bigger possibility our bot will raise or even play all in.
3.2. Flop, Turn, The River

Our bot play strategy at the flop, the turn and the river is always the same and in fact pretty simple. All we have to do before make a decision is to evaluate our probability of winning, drawing and loosing that deal, with actual cards configuration. So the best way to do it is randomizing a lot of times (we are making it a 1000 times) a further part of that deal, so we can see our percentage probability to win. Our bot has also a constant “tightness” ratio which can take values between 0 and 1. The bigger it is, the less risk our bot is taking while making decision. And now depending on probability of win, that we have already evaluated, and a tightness ratio, bot is making decision, how to play, and how much is the maximum bet that it can raise or call. If the maximum bet is greater than actual bet on the table bot rises or calls, otherwise it folds.

3.3. Implementation

The bot has been implemented in Java language. A brain of our poker bot is PokerCalculator class which is implementing methods that can evaluate probability of winning, maximum bet that bot can actually play and finally take a decision. There are two of them:
• **public static** CalcResult calculateAtFlop(Card c1, Card c2, List<Card> tableCards, **int** opponents, **int** samples) – calculates winning, drawing and loosing probability. C1 and c2 are cards on our hand, tableCards is a list of cards that are on the table. We are also using number of opponents and number of sampling (we use 1000 as I written before).

• **public static** BotDecision getDecision(...) – getting a bot decision.

4. **Comparison with other bots**

We have compared our bot with three other bots: Random bot, Call bot and Fold bot.

<table>
<thead>
<tr>
<th>Bot name</th>
<th>Game played</th>
<th>Wins</th>
<th>Lost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Call bot</td>
<td>100</td>
<td>78</td>
<td>22</td>
</tr>
<tr>
<td>Random bot</td>
<td>100</td>
<td>82</td>
<td>18</td>
</tr>
<tr>
<td>Fold bot</td>
<td>100</td>
<td>92</td>
<td>8</td>
</tr>
</tbody>
</table>