A study of Poker bots and the existent models behaviour

Andreas Mihaloianis

3 May 2017

1 Abstract

2 Introduction

3 Background and context

3.1 Overview

Poker is the world's most popular card game and one of the most popular sports in the world. According to some sources, more than 100 million people in the world enjoy playing poker on a regular basis [1]. It combines strategic challenges with psychological elements. With poker, you don't play just your cards - you also need to analyse and react to your opponents. Of course, having a computational environment makes this difficult, as it is still computationally impossible to assess live tells and behaviour. Because of this, poker AI robots need to have an extremely strong mathematical background and build strategies on top of that in order to become successful while playing the game.

The most played variant of poker is Texas Hold'em [2]. Texas Hold'em is one of the most "board" games of poker. In Texas Hold'em, each player receives 2 hole cards, which are considered personal. Then, 5 cards are drawn on the table. These cards are called the "community cards". The player with the best 5-card combination chosen from within the hole cards and the board cards wins the money in that round. Also, players are allowed to bet while the board is being drawn. There are 4 rounds of bets in total, with each round giving the opportunity to all the players to bet, in turn. In this way, players with a weaker hand are being given the chance to win a hand, by betting their opponents out of the game. The full rules of Texas Hold'em can be found here: https://en.wikipedia.org/wiki/Texas_hold_'em [3].

There is a general controversy around the game of Texas Hold'em which is present for most of the other poker games, too: Is poker a game of luck or skill? Many people believe that, since the cards are distributed randomly at the start of each hand, poker has luck as its main feature. Even Phil Helmuth, a well-known profesional poker player and 11-times WSOP(World Series of Poker) Bracelet winner, once said "If it weren't for luck, I'd win every time." [4]. So, there certainly is reason to believe that luck is the main feature of poker, but is it true? First of all, there is no doubt in the fact that luck is part of the game. On the other hand, there are many experienced players, including Phil Helmuth himself, that have won tournaments on several occasions. So, if poker would have been a pure luck game, this wouldn't happen, as new "lucky" winners would emerge almost every tournament. Secondly, the structure of the game enables skilled individuals to emerge as winners, by making better, more profitable plays than an average player. This might include bluffing an opponent, getting value out of a better hand or folding when having a weaker holding. Therefore, skill is an absolute must in order to play a game of poker efficiently. Most of the good players use their skill to get in a advantageous situation, such as a 70%-30% favourite. From there on, it's all about luck, but on a long run, the players who are able to get themselves in more such spots are the ones who win. In conclusion, poker is a game of skill in the first place, with luck only making it more appealing because of the uncertainty that it is unveiled in.

Poker games are traditionally played by 2-9 individuals sitting at a table and betting either chips or money. As technology evolved, so did the opportunity of playing poker. Nowadays, 1 in 4 dollars are gambled online [5], with poker being one of the most successful casino-related businesses, summing up to a grand total of $33 billion dollar market [6]. After poker games have started to move online, poker-playing bots started emerging in the field. Many researchers have since tried different approaches in order to solve the computationally difficult game of Texas Hold'em. In the following part of the "Background and context" section I will present the most honourable of those, continuing with presenting my own view of how the game-solving algorithms should be configured like in the next sections.

3.2 A generic view on poker bots

There are multiple ways of constructing poker bots. One can think of a mathematical approach, while others can try machine learning and AI techniques in order to solve the game of poker. Either way, since Poker is a game of uncertainty, even newest technologies seem to be far away from being able to virtually win all the time. Moreover, because no-limit Hold'em is a game where a vast number of parameters has to be taken into account, poker bot specialists have trouble even in finding methods of being more successful than an average professional player, mostly because of the fact that a human player can also take information out of emotions and physical tells during a Poker game.

3.2.1 Borel and von Neumann Poker Models

The most low-level Poker model is the Borel Poker model, explained in a detailed way by Émile Borel in his book "Applications de la théorie des probabilités aux Jeux de Hasard" [7]. A very similar model was proposed by John von Neumann in his book "Theory of Games and Economic Behavior" [8]. The two forms of poker represent a starting point for analysing poker as a game of improbabilities. The Poker chapter in the "Fun and Games, A text on game theory" book, by Binmore [9] presents the knowledge of game theory applied on these simple forms of Poker. In these simplistic variants of Poker, the two players each receive a random "card", drawn from a uniform distribution on [0,1]. Each of the players knows their own card, but cannot find out what the opponent's card is. Before the cards are dealt, both of them have to post a unit of value, so that if one of them folds instantaneously, the other one has something to win. Therefore, for each decision someone makes, there is a payoff that needs to be taken into account. After that, the first player has to make a decision, having known his cards: fold/check or bet (add b units of value to the pot). The second player can call or fold afterwards, if the first player hansn't folded. If no player has folded, then the player with the biggest card value wins the pot. The only difference between the two models is that in Borel's model, the first player can only fold or raise, while in von Neumann's he can only check or raise. While this game might sound simplistic for poker players, it contains a strategic and mathematical background that can help deal with other variants of poker, too. Borel has solved [1] this game by using a strategy called "la Relance" [10], also described in his book. This method consists of creating a betting tree for the described game, in a similar way betting trees are being constructed for any game theory problem.

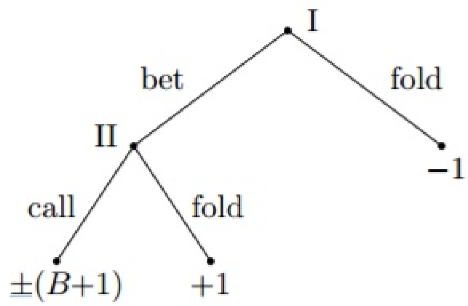

Figure 1: Borel's La Relance betting tree

In the betting tree for "la Relance" [Figure 1], the value indicated at the leafs of the tree represent the winnings from Player I's perspective. The ± sign represents the uncertainty regarding the win/loss of the respective hand which will be determined by the card values of the two players. In the analysis, Borel found out that there is a unique optimal strategy for Player II. His strategy has to be on the form of some number c, c ∈ [0,1]. He concluded that the unique optimal stragegy for Player II is[11]:

- Call if [2]Y > c
- Fold if Y < c

, where c = B/(B+2). On the other hand, Player I's strategy is not unique and can be though of at different levels, since he has, on top of calculating Nash equilibrium, the opportunity to also bluff in some situations. Even so, Borel found all of them. The common feature of all possible Player I's strategies is his betting threshold. More precisely, he should bet if [3]X > c2. Also, in order for an strategy to be optimal, Player I has to bet with a 1-c proportion for all the hands on which X<c, and fold a c proportion of those. As a conclusion, Borel outlined the value for his "la Relance" poker model as:

$$V(B) = -\frac{B^2}{(B+2)^2} \tag{1}$$

Because this value is negative, we can conclude that this game advantages Player II. Realising this, von Neumann created a model which gives Player I more advantage. Therefore, if he elects not to bet, he still stands chances of winning, when X > Y. As a result, von Neumann's betting tree looks is presented in Figure 2. This modification helps modeling the bluffing concept in more detail, because if Player I elects to bluff with an average hand and looses, will loose B+1, while if he chooses not to bluff, his expected winnings are 0 (± 1).

[1]Solving a (Poker) game means discovering a strategy that is optimal for every possible holding for a player.
[2]Y is the value distributed from [0,1] for Player II
[3]X is the value distributed from [0,1] for Player II

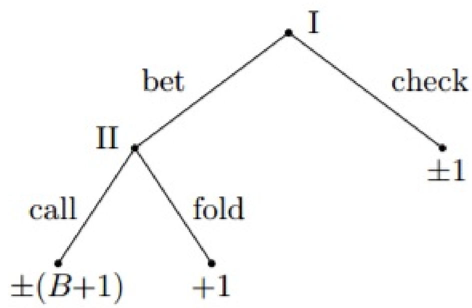

Figure 2: von Neumann's betting tree

The strategy for Player I changes in this situation. As a result, Player I only has one (unique) strategy now, which is in terms of a and b, two values from [0,1], with a<b. The reason for having 2 thresholds instead of one is because even though Player I was only 2 possible ways of playing the hand, a bet decision might come out of a pure value extraction attempt, or a bluff attempt. Therefore, Player I should bet if X>b(for value) or X<a(bluff) and check otherwise. On the other hand, Player II only has one threshold, namely c. He should call if Y>c and fold otherwise. It turns out the following equation holds, as visible in the von Neumann's optimal play decision figure (Figure 3).

$$0 < a < c < b < 1 \tag{2}$$

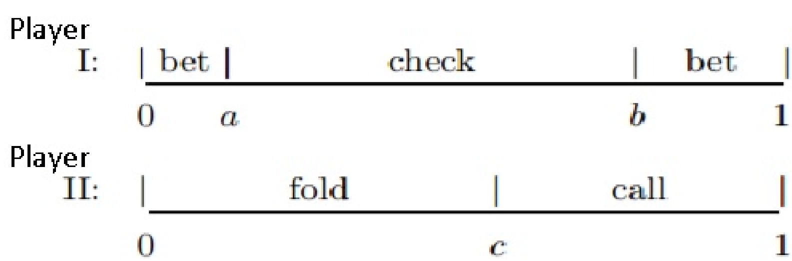

Figure 3: von Neumann's optimal play visualisation

Von Neumann concluded that the expected value for his poker model is:

$$V(B) = \frac{B}{(B+1)(B+4)} \tag{3}$$

The fact that von Neumann's model advantages first player in comparison to Borel's can be seen from the formula. V(B) is a positive term, so the previous statement stands. Also, for completeness, von Neumann computed the values of a, b and c from within the optimal play formulas. These are:

$$
\begin{aligned}
a &= \frac{B}{(B+1)(B+4)} \\
b &= \frac{B^2+4B+2}{(B+1)(B+4)} \\
c &= \frac{B(B+3)}{(B+1)(B+4)}
\end{aligned} \tag{4}
$$

3.2.2 Heads-up Limit Hold'em is solved

While Borel and von Neumann models might have the most low-level basis for poker game theory, both are far from replicating a true poker game. Poker consists of two or more players receiving two cards each (rather then a random number) and betting value (chips/money) according to their holdings, the community cards and the read the player might have on the opponent's holding(s). First, let's consider the most simple situation, the head's up game. A head's up game is, as simple as it can get, a two-players poker game. While this might still sound simplistic compared to a full-ring poker game, its complexity is huge. Some of the complexity goes down to the fact that when a players raises with 10 or 20 chips, the two are considered different moves even though the action was a raise in both cases. In order to restrict such behaviour, a Limit variant of the game has been developed. In this variant, if a player decides to raise [13], he can only do it by a designated amount, which is proportional to the big blind. In such fashion, the complexity of the game is trimmed down. Because of its reduced complexity compared to normal Hold'em, poker AI specialists have tried to produce bots that play HULHE (Head's Up Limit Hold'Em). Still, the complexity of this game is astonishing. According to Alberta Poker Research Group [12], there are $3,19*10^{14}$ information sets involved in HULHE, with $1,38*10^{13}$ sets after removing game symmetries. In order to deal with this much load of information, the biggest challenges that occurred were memory and computation. In order to deal with these issues, the Alberta Poker Research Group came up with a CFR^{+} extension of their previous CFR algorithm that handles data at this kind of scale.

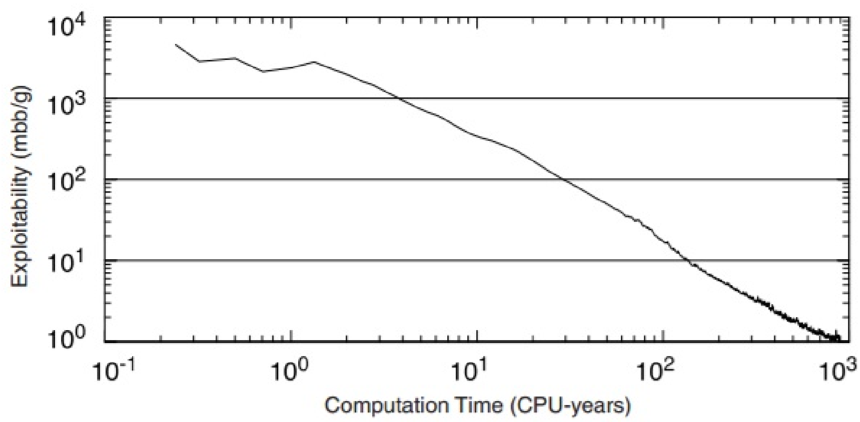

Figure 4: Exploitability of the approximate solution with increasing computation (measured in milli-big-blinds per game (mbb/g))

CFR (Counterfactual Regret Minimization) [15] is an iterative method of approximating a Nash equilibrium in a game with incomplete information (like Poker). The method of achieving this approximation is through the process of repeated self-play between two regret-minimizing algorithms, where regret is described as being the loss of a strategy/algorithm generated by a certain decision which might not have been in accordance to the single best deterministic strategy. The key of CFR is that instead of storing and minimizing the regret over the computationally huge number of different possible strategies, it uses a modified regret for each information set and subsequent action, which can be used to form an upper bound on the regret for any deterministic strategy. Therefore, the Nash Equilibrium is retrieved by averaging each player's strategies over all of the iterations. Obviously, as the number of iterations gets bigger, so does the quality of the approximation [14]. The quality of this approximation can most efficiently be measured using the exploitability of the chosen strategy. As you can see in figure 4, the exploitability levels go down as the number of iterations goes up.

While the memory issue of the previously-described CFR algorithm can be solved with better hardware or clustering, CFR^+ needs to also cover the computation gap in order to provide a better solution. As a result, in addition to its previous variant, CFR^+ does exhaustive iterations over the entire game tree and uses a variant of regret matching where regrets are non-negative. Another particular aspect of CFR^+ is the fact that after a certain time, the regret of players' current strategies regularly approaches zero. As a result, when calculating the overall CFR^+ solution, the step of computing and storing the average strategy can be skipped, because using the player's current strategies is much faster computationally.

In order to visually assess the importance and power of the CFR^+ algorithm, one can refer to figure 5 and realise that the last step made by the people from the Alberta University of Poker is a huge one.

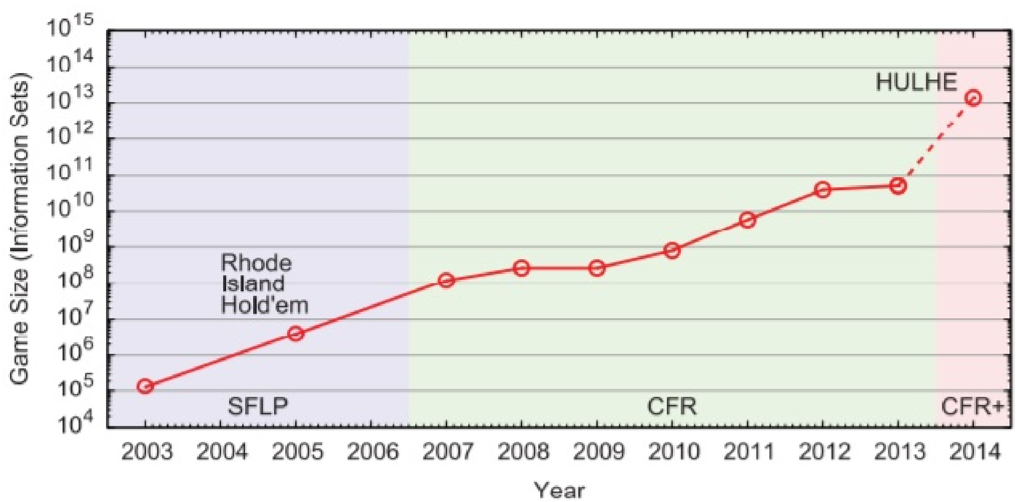

Figure 5: Increasing sizes of imperfect-information games solved over time measured in unique information sets (i.e., after symmetries are removed).The shaded regions refer to the technique used to achieve the result; the dashed line shows the result established from the CFR to the CFR^+ model.

The CFR^+ model is of such importance that its usefulness is not restricted to poker. Poker is just a testbed for such technologies and all environments that have uncertainty within can apply such techniques for solving different trade-offs. For instance, CFR algorithms have been used in game-theoretic applications involving security, including complex systems for airport checkpoints [16].

3.2.3 Heads-up No-Limit Hold'em AIs

As previously mentioned, research in the poker bots domain tends to follow the heads-up variant, simply because this is the least complex forms of poker. Lately, a couple of years after the Limit Heads-up game has been solved by poker AI specialists, another spark of success enlightened the AI specialists. 2017 marks the appearance of 2 heads-up poker AIs that have played the game at least at the same level as ones of the best human heads-up specialists [17]. The 2 bots are called DeepStack and Libratus and have been developed by the Alberta Poker Group and Carnegie Mellon University in Pennsylvania respectively. In order to better understand the complexity difference between HULHE and HUNHE(Heads-up No-Limit Hold'em), we just need to compare the information sets that are covered by the 2 games. That is, 10^{160}(for no-limit) vs. 10^{14}(for limit)[17]. One of the ways of dealing with this is by bucketing and trimming down all the possibilities to a computable complexity. DeepStack, for intance, plays many games against itself in order to "squeeze" the complexity down to the computable value of 10^{14}.

There are a couple of things that stand out in terms of similarity of these 2 poker bots. First of all, they have both proven to be extremely successful against top human players [18] and they were both released in the beginning months of 2017. Because of that, not surprisingly, they also share some technology approaches. For example, both use a CFR model for solving their endgames, which proves once more its power and usefulness in an environment described by uncertain events. Secondly, both of the approaches try to approximate a Nash equilibrium in order to decide on what the best strategy would be [19] [20]. This is an expected behaviour for a heads-up game, though, as it describes a two-player zero-sum perfect-recall, which can be best resolved using Nash Equilibrium.

The research in these area will certainly continue and even though this high milestone of beating human players has been reached by both of the AIs, there are certain aspects that can be improved in the computer-decided strategies that these bots follow. That is exactly the reason why Libratus played against 4 of the top No-Limit Heads-up specialists the second time [21], completing 120000 hands in late January 2017. In his first attempt, Libratus failed to win, but on this occasion he triumphed by a significant statistical margin, overtaking the best-performing human player by $85,649 overall, proving once again that CFR^+ and its variations can be the winning choice in such environments.

3.2.4 Full Ring Poker

// POKI

4 Design and Implementation

5 Evaluation and Results

6 Conclusions

References

[1] https://www.pokerstrategy.com/poker-the-game/

[2] https://www.cardschat.com/top-10-poker-games.php

[3] https://en.wikipedia.org/wiki/Texas_hold_'em

[4] http://realmoney.durrrrchallenge.com/skill-luck-paradox/

[5] http://www.indian.senate.gov/sites/default/files/upload/files/old_hearings/GrantEveTestimony111711pageid=9698.pdf

[6] http://www.uspokersite.net/online-poker-real-money/

[7] Félix Édouard Justin Émile Borel - "Applications de la théorie des probabilités aux Jeux de Hasard" (1938)

[8] John von Neumann and Morgenstern - "Theory of Games and Economic Behavior" (1944)

[9] K.Binmore - "Fun and Games, A text on game theory" (1992)

[10] http://www.chrissnijders.com/eth/Ferguson.pdf

[11] http://www.cs.cmu.edu/afs/cs/academic/class/15859-f01/www/notes/mat.pdf

[12] http://poker.cs.ualberta.ca/

[13] http://www.pokerlistings.com/texas-holdem-betting-rules

[14] http://science.sciencemag.org/content/347/6218/145

[15] http://poker.cs.ualberta.ca/publications/NIPS07-cfr.pdf

[16] http://ai.cs.unibas.ch/_files/teaching/fs15/ki/material/ki02-poker.pdf

[17] https://www.theregister.co.uk/2017/01/14/deepstack/

[18] http://www.theglobeandmail.com/news/national/artificial-intelligence-crosses-the-poker-barrier/article34185354/

[19] https://arxiv.org/pdf/1701.01724.pdf

[20] https://arxiv.org/pdf/1702.04849.pdf

[21] https://www.riverscasino.com/pittsburgh/BrainsVsAI/