Poker’s Role in the Rise of AI

Zackary Medeiros

Introduction

Gambling has a certain alure around it. You have the potential to get rich quick, all the while rebelling against society and many religions as you dip your toes into the taboo. Of all the games in a casino, Poker has to be the most interesting. It requires the most skill of any game, and it’s one of the only games where you are not playing against the house, instead, you are playing against only other players. Poker is also very popular outside of casinos, as it is a fun game to play with friends. As with any game, especially one where you have the chance to make a lot of money, people want to get good at it. With the development of computers over the last seventy-five years or so, AI or ‘bots’ have been developed to play games. These bots were created to give single players someone to play with, and eventually to create a challenge for even the best players in the world.

One of the most popular games of all time, Chess, is often at the center stage of AI development. This is for many good reasons, as chess is a simple game, but becoming very good is very hard, which is why chess grandmasters have their names recognized around the world. Since the beginning of computing in the ‘50s, computer scientists have tried to make computers that could play chess. There were working models all the way at the beginning, however, they were easily beaten by human players. Chess AI did not become good until the ‘90s. (Ashe) Teaching AI how to play chess makes sense, since it is a “perfect information” game. A perfect information game is a game in which every player knows all the information, in other words there is no hidden information, both players know the same things. Perfect information games like chess have optimal moves and strategies, that can be exploited by a computer, since humans cannot think more than a few moves into the future, and they can always miss something.

Perfect information games like checkers (1995), chess (1997), and go (2017) have all been considered to be solved, since the AI can successfully beat humans, basically every time. [Solving games will be discussed further later in this paper.] But not every game is a Perfect information game. Poker is in the other category, namely “hidden information” games. In any version of poker, there is at least some information that only a single player knows, in Texas Hold’em, the most popular poker game, and the one this paper will be focused on, each player has two ‘hole’ cards, which are only known by the individual player. There are also five community cards, which all players can see. Hidden information makes creating AI bots harder, since you cannot know the true optimal move to make. Player behavior is also an important part of poker. Bluffing, player tells, and patterns of play make reading your opponent(s) extremely important to winning.

Hidden information games also mirror real life more than perfect information games, since it is rare that you will know all the information in any real-life scenario. Therefore, successfully creating AI that can perform well in hidden information games has a high incentive, because the AI may then be able to be tailored to real life scenarios. Hidden information AIs have been found to be highly applicable in business strategy, negotiation, strategic pricing, finance, cybersecurity, military applications, and more. They can even take up less computing resources than their perfect information counterparts. This, and the lure of becoming rich, spurred the development of poker AI.

Bot Development

We will be focusing on the No-Limit Texas Hold'em game of poker, and use of the term 'poker' will be synonymous with NLTH, unless otherwise specified. Before AIs were developed for poker, the idea of a poker solver was floating around the minds of top players. This solver would have the mathematically optimal, unbeatable strategy to one-on-one NLTH, known as Nash Equilibrium. Nash Equilibrium is a concept in game theory where no player could gain by changing their own strategy. Basically, if the solver could attain Nash Equilibrium, then it would be unbeatable, since the opponent cannot get any better. Proving the existence of Nash Equilibrium in Head's up NLTH is relatively straightforward, however actually finding it would prove challenging.

A poker player and programmer named Oleg Ostroumov was the first person to develop a Poker Solver, back in 2013. He became passionate about poker after being introduced to the game by a professional player he knew. Once he was hooked, he set his eyes on creating a solver. He eventually created a program that calculated an approximation of the Nash Equilibrium starting from the flop [when the first three community cards are revealed], with a limited number of bet sizes. His program played against itself, starting randomly, then utilizing the Monte-Carlo Counterfactual Regret Minimization algorithm to get better. [It is common in AI neural network training for AI to repeatably play itself and/or a game to get better, however, nowadays, we have discovered better ways to create poker bots.] Ostroumov had successfully created the world's first poker solver, which could be used to enhance the skills of all players, even those at the top of the game. One of the ways he knew he had cracked the code was the fact that his solver made some unorthodox moves, considered against the 'book'.

"*In 2012, there was a unanimous consensus among poker players that after a check-call on the flop, one should always check the turn. This strategy is correct… most of the time. However, the solver bet 11% of the time after a check-call on the flop, valuebetting trips of 9 and semibluffing flush draws — a highly unorthodox move in this context. Trueteller [a poker professional Ostroumov was trying to sell a license to] suspected this was the correct strategy, but no one played like that. He also knew that I couldn't have hardcoded it or overfitted the algorithm to produce this outcome.*" (Ostroumov)

Chess bots have been found to do similar things, going against human chess principles such as pushing pawns in front of their king, and weakening an area of the board where they cannot claim any advantage. (GothamChess) These kinds of moves may seem confusing to us humans, but they make sense to the AI, since it is looking far further into the future, and far deeper into the strategy of the game.

Following Ostroumov's solver, other solvers were developed, and then AI bots were created. In 2017, researchers at Carnegie Melon University developed Libratus, an AI that, in their mind, 'solved' Head's Up No Limit Texas Hold'em. Instead of creating a bot through neural networks and self-play, they used a three-pronged approach which combined "an algorithm for computing a blueprint for the overall strategy, an algorithm that fleshes out the details of the strategy for subgames that are reached during play, and a self-improver algorithm that fixes potential weaknesses that opponents have identified in the blueprint strategy." (Brown) These three algorithms came together to create an unbeatable bot. The first prong builds a blueprint strategy for the game overall, to get around the issue of imperfect information. In this algorithm, a game tree is created with all the possible moves and results of those moves, and the paths are analyzed to find the best move forward. In creating this tree, the game is generalized to make it simpler and therefore the make tree smaller and more manageable. For instance, the bet sizes are cut down, because a $100 bet is not very different from a $101 bet. Libratus uses these abstractions in the first two rounds of play, before it switches to the second prong in the last two rounds.

The second prong of Libratus takes it to the next level. When reaching a point in the game where the remaining game tree is relatively small, Libratus constructs a much more detailed abstraction for the remaining subgame and solves it in real time. The problem with this approach is that a subgame cannot be solved on its own because the optimal strategy may rely on other subgames being solved. One way to get around this is to assume the opponent is using the blueprint strategy that was created in the first part of the approach, this however is open to exploits. If an opponent knew Libratus was assuming this, they could simply play with a different strategy and Libratus would then not find the optimal subgame. Instead, Libratus uses 'safe subgame solving', where it guarantees that the new strategy leaves the opponent no better than it was before no matter how the opponent reacts. The strategy is described below:

*"Let σ_i be a strategy for a two-player zero-sum perfect-recall game, let S be a set of nonoverlapping subgames in the game, and let σ_i^* be the least-exploitable strategy that differs from σ_i only in S. Assume that for any opponent decision point (a hand in the case of poker) and any subgame in S, our estimate of the opponent's value in a best response to σ_i^* for that decision point in that subgame is off by at most Δ. Applying estimated-Maxmargin subgame solving to any subgame in S reached during play results in overall exploitability at most 2Δ higher than that of σ_i^*." (*Brown*)*

In short, Libratus expects that its opponent is using an optimal strategy, and then it calculates the expected value of all the subgames, where the opponent has a random 2 card poker hand, and Libratus has its hand. Based on the expected value of the subgames, and what the algorithm believes the opponent will do, Libratus decides the next move to make. And then, the cycle repeats. Safe sub-game solving has been around for a while, but it was never more effective than just relying on the blueprint, until Libratus made some improvements to subgame solving. For example, Libratus uses estimated values in subgame solving rather than upper bounds on values, which helps strategies be more realistic and not overly conservative. Libratus also cuts out unnecessary information, like subgames where it has 7 2 off suit, the worst hand in poker, which would have been immediately folded, thus never arriving in a subgame. Another thing the algorithm does when needed is calculate in real time augmented subgames when the opponent makes a move that is off of the abstraction tree, which helps keep the abstraction relatively small.

The final prong is self-improvement. Libratus enhances the blueprint strategy in the background as the game goes on. Basically, it fills in missing branches where relevant, informed by the actions of the opponent. Calculating the whole game tree in advance is impossible, so this strategy allows for the tree to be more detailed (less abstracted) on the paths the game is actually using. Once developed, Libratus was tested against the top poker bots of the time, and it beat all of them. The model was so good, that there was no need to analyze players for tells, and any bluffing was caught by the bot's strategy. Different versions of Libratus were tested against the bots, with each version adding on one of the prongs, or some other improvement. Libratus defeated all of them, even at its lowest level where only the blueprint was used. With its three prongs, Libratus became the first AI poker bot to beat professional players. In fact, it did not just beat professional players on occasion, it beat each of them practically every time.

"In January 2017, Libratus played against a team of four top HUNL-specialist professionals in a 120,000-hand Brains vs. Artificial Intelligence: Upping the Ante challenge match over 20 days. The participants were Jason Les, Dong Kim, Daniel McCauley, and Jimmy Chou. A prize pool of $200,000 was allocated to the four humans in aggregate. Each human was guaranteed $20,000 of that pool. The remaining $120,000 was divided among them on the basis of how much better the human did against Libratus than the worst-performing of the four humans. Libratus decisively defeated the humans by a margin of 147 mbb/game, with 99.98% statistical significance and a P value of 0.0002 (if the hands are treated as independent and identically distributed). It also beat each of the humans individually." (Brown)

Solving the Game

With the creation of Libratus, many believe that Head's Up, No Limit Texas Hold'em has been solved. When we talk about something being solved, or proven, there are different definitions depending on the field of study. In Mathematics, and therefore computer science, we focus on rigorous proof. These proofs should be based on a form of logic, which could be Boolean logic, propositional logic, inductive logic, Hoare logic, or something else. In any case, the proof should be built with concepts or facts that are true, and in the end the proof should not be able to be proven false. In science, something being ‘solved’ is similar, in that we base our solutions/proofs on facts and real-world evidence, however, in science nothing is really truly solved, since science allows for things to be disproven, which is why discoveries are labeled as theories. When we say that HUNLTH has been solved, we are taking ideas from both Mathematics and Science. The algorithms in Libratus are built up using Mathematical ideas, and they are believed to be correct. For example, we believe Liberatus does use the most optimal strategy, and thus it has solved the game. We believe this because Liberatus plays making sure that no strategy can do better than it, therefore a better strategy cannot exist. However, actually rigorously proving this is difficult. Scientific ideas come in when we look at the results of running experiments using Liberatus. The researchers at Carnegie Mellon put Libratus up against the other top poker bots, and it decisively beat all of them with relative ease. Professional poker players were no match for it either, as all of them were beaten as well, and by large margins. This is why people believe HUNLTH is solved: we have a tool that can reliably win the game every time. People believe that chess, checkers, and go are solved because of similar logic. Each of the games has an unbeatable superhuman AI, and there is no reason to believe that anything will cause the AI to lose its status.

Now that HUNLTH was solved, or at least had a near optimal solution, the next logical step was to move on to multiplayer No Limit Texas Hold’em. Historically, AI bots have almost exclusively been developed to play 1-on-1 games, or games with two teams. This is because creating algorithms for use against one opponent is far simpler than trying to take into account a variable number of opponents. Also, in a hidden information game like poker, it is hard enough to create a strategy not knowing the opponent's hand, but working with many opponents with unknown hands makes things exponentially more complicated. NLTH is the most popular and most played poker format in the world, so these challenges were accepted by the researchers at Carnegie Mellon.

Their new AI Pluribus was built off several innovations on Libratus. One of the most important innovations that Pluribus has is that it uses a new search algorithm to evaluate its options more efficiently by using a depth limited search instead of searching to the end of the game tree. (Meta) This allows the AI to still be effective against more than one opponent, since even with more unknown information, it is faster than the old model. This model also made Pluribus use less resources than Libratus. Pluribus was tested in two major NLTH experiments. The first was in a game of six players, where there were five human professionals, and Pluribus was the sixth player. The other experiment was a game where there were five AI players, and one human professional. In both experiments, the AI defeated the humans, less decisively than Libratus, but still clear enough to argue that NLTH had been solved as well. Solving a game like NLTH would be even more significant than the heads-up version. No game of this style had been perfected by AI yet, and real-world scenarios are better represented by a multiplayer hidden information game. Cybersecurity, traffic, online auctions, and many more real-life applications would benefit from a tool that could accurately solve their situations, situations that have many factors and lots of hidden information. Solving NLTH would be one of the greatest advancements in AI so far and would further interest and development into AI.

Conclusion

Gambling is an intriguing activity, that is very alluring to people due to the chance of large fortune. Poker is the most skillful casino game, which can also be enjoyed at home with friends. AI bots have been created to beat popular games such as chess, checkers, and go. Poker was a logical step for its popularity, and the challenge of a hidden information game. Before AI bots were developed, solvers for various versions of the game were created. Then, researchers at Carnegie Mellon University created Libratus, a bot that plays Heads Up No Limit Texas Hold'em. Libratus was able to beat professional poker players, and every other bot that had been created up to that point. The AI was so good, that it is believed to have solved HUNLTH. After Libratus, Pluribus was created, which improved upon Libratus, and could play No Limit Texas Hold'em with six players. Pluribus also beat the pros and may have solved NLTH. These developments have made huge leaps in AI research and applications. Hidden information games, especially those with more than two players, more accurately reflect real life than perfect information games. The work done on these bots can be used to help apply AI to real life scenarios in business, cybersecurity, the military, and many more fields.

Works Cited

AI Playing Games. Directed by Jabril Ashe, vol. 12, 2019, https://www.youtube.com/watch?v=nw7zmdBLQ6U.

Brown, Noam, and Tuomas Sandholm. "Superhuman AI for Heads-up No-Limit Poker: Libratus Beats Top Professionals." Science, vol. 359, no. 6374, Jan. 2018, pp. 418–24. DOI.org (Crossref), https://doi.org/10.1126/science.aao1733.

---. "Superhuman AI for Multiplayer Poker." Science, vol. 365, no. 6456, Aug. 2019, pp. 885–90. DOI.org (Crossref), https://doi.org/10.1126/science.aay2400.

"Computer Poker Player." Wikipedia, 21 Nov. 2024. Wikipedia, https://en.wikipedia.org/w/index.php?title=Computer_poker_player&oldid=1258705675.

Meta. “Facebook, Carnegie Mellon Build First AI That Beats Pros in 6-Player Poker.” Meta Research, 11 July 2019, https://ai.meta.com/blog/pluribus-first-ai-to-beat-pros-in-6-player-poker/.

“Nash Equilibrium.” Wikipedia, 24 Oct. 2024. Wikipedia, https://en.wikipedia.org/w/index.php?title=Nash_equilibrium&oldid=1253087264.

Ostroumov, Oleg. “How I Created a Poker Solver and Made $500k at the Age of 23.” Medium, 24 Dec. 2023, https://medium.com/@olegostroumov/worlds-first-poker-solver-6b1dbe80d0ee.

“Pluribus (Poker Bot).” Wikipedia, 5 Jan. 2024. Wikipedia, https://en.wikipedia.org/w/index.php?title=Pluribus_(poker_bot)&oldid=1193778294.

Poker’s Greatest Tournament Run. Directed by EmpLemon, 2024, https://www.youtube.com/watch?v=xXJM9RtbNNs.

Statt, Nick. “DeepMind’s StarCraft 2 AI Is Now Better than 99.8 Percent of All Human Players.” The Verge, 30 Oct. 2019, https://www.theverge.com/2019/10/30/20939147/deepmind-google-alphastar-starcraft-2-research-grandmaster-level.

---. “OpenAI’s Dota 2 AI Steamrolls World Champion e-Sports Team with Back-to-Back Victories.” The Verge, 13 Apr. 2019, https://www.theverge.com/2019/4/13/18309459/openai-five-dota-2-finals-ai-bot-competition-og-e-sports-the-international-champion.

“Stockfish (Chess).” Wikipedia, 21 Nov. 2024. Wikipedia, https://en.wikipedia.org/w/index.php?title=Stockfish_(chess)&oldid=1258713963.

Vernon, Hayden. “I Turned £1 Into £220 with the Help of an Online Poker Bot.” VICE, 26 June 2020, https://www.vice.com/en/article/online-poker-bot-review-winnings/.

Why AI Chess Bots Are Virtually Unbeatable. Directed by GothamChess, 2023, https://www.youtube.com/watch?v=CdFLEfRr3Qk.