Poker Agent

Tomer Dobkin

May 23, 2020

1 Introduction

the game Texas Hold'm Poker is a popular game, and it is also has vast of researches that their goal is to create good computer programs that play poker, we will use the term - poker agents.
A strong motivation to this researches is the fact that poker models are interesting use case of AI. the use case is a game (well defined) with imperfect information (about the world) and element of chance.

1.1 Rules

as reviewed in Computer Poker - A review (Rubin & Watson, 2011).
The game of Texas Hold'em is played in 4 stages – preflop, flop, turn and river. During the preflop all players at the table are dealt two hole cards, which only they can see. Before any betting takes place, two forced bets are contributed to the pot, i.e. the small blind and the big blind. The big blind is typically double that of the small blind. The player to the left of the big blind, known as under the gun, then begins the betting by either folding, calling or raising. The possible betting actions common to all variations of poker are described as follows:
Fold: When a player contributes no further chips to the pot and abandons their hand and any right to contest the chips that have been added to the pot.
Check/Call: When a player commits the minimum amount of chips possible in order to stay in the hand and continue to contest the pot. A check requires a commitment of zero further chips, whereas a call requires an amount greater than zero.
Bet/Raise: When a player commits greater than the minimum amount of chips necessary to stay in the hand. When the player could have checked, but decides to invest further chips in the pot, this is known as a bet. When the player could have called a bet, but decides to invest further chips in the pot, this is known as a raise. In a limit game all bets are in increments of a certain amount. In a no limit game players can wager up to the total amount of chips they possess in front of them. Once the betting is complete, as long as at least two players still remain in the hand, play continues on to the next stage. Each further stage involves the drawing of community cards from the shuffled deck of

cards as follows: flop – 3 community cards, turn – 1 community card, river – 1 community card. During each stage players combine their hole cards with the public community cards to form their best 5 card poker hand. Each stage also involves its own round of betting and play continues as long as there are players left who have not folded their hands. A showdown occurs after the river where the remaining players reveal their hole cards and the player with the best hand wins all the chips in the pot. If two or more players have the same best hand then the pot is split amongst the winners. (TODO - expand what is "the best hand")

2 Literature Review

The researches of poker are really rich and started way from (TODO complete) and are last to our days. their are competition of poker agents, the most famous one is Association for the Advancement of Artificial Intelligence (AAAI) Computer Poker Competition. the research are used of there from variety of subjects from math and computer science:

1. AI
2. Game Theory

the first agent that won a human, a common comparison that is done when dealing with AI that play games (like with Chess and Deep Blue) , (TODO complete)

as mentioned in Computer Poker - A review, there are several approaches that are used to create poker agent

2.1 Knowledge based agents

In this approach the programmer encodes expert knowledge to set of "if-else" statements, the disadvantage of this approach is that it's not so hard to human or other agent to learn the knowledge base and use it to establish winning strategy. another disadvantage it's that in practice it is not perform very well, maybe due to the fact that it's hard to encode all the expertise, and the knowledge of experts is also bounded.

2.2 Formula based agents

similar to knowledge base , in this approach the agent will act from evaluation of heuristic function that believed to maximize the expectation. This approach tackle similar disadvantages as in the knwledge base approach. (TODO - add examples of formulas)

2.3 Simulation Driven

Another classical approach is to use MCMC simulation to calculate expectation of given hole and community cards and to play with moves that maximize this expectation. the disadvantage of this is the computed simulation did'nt give us usually the best move (TODO - expand and explain why). Due to the fact that the game is can be models by a tree , each player turn is a node and each possible move is a branch, we can use MCST to simulate the game. (TODO - expand more on MCST)

2.4 Game theoretic equilibrium solutions

Poker is a game played with number of players. we can look at this game as a tree of moves, with branch factor as the number of legal moves at each stage. If we assume that the other player play perfect game (Game theory wise) we can use search algorithm, simple min-max will not work here due to the fact that each play has imperfect information, the fact that the game is includes element of change is dealth with the algorithm (expectimax TODO-add reference and expend). we can try other algorithms. Moreover , the number of possible states is really high so game search algorithm will struggle to find an optimum, so an approach that is tried in the past is to reduce the number of states by abstraction, we can see it as equivalent classes, some states of the original game will map to the same EC in the simplified game. and the algorithms will run on the simplified version. (TODO expand on lossless and lossy abstraction) The main disadvantage of this approach is the assumption of perfect play , so in the long run because it's near-Nash equilibrium solution , in practice the agent misses chances to exploit other players when they don't play perfect strategy. another technique that used elaborately in DeepStack (Moravčík et al., 2017) is Counterfactual Regret Minimisation (CFR) (TODO - extend)

2.5 Exploitive counter-strategies

An interesting approach is to use exploitative counter-strategies , the understanding that to achieve Nash equilibrium solution is not realistic , our opponent can reach only near-Nash equilibrium solution, make the opponent "vulnerable" to counter-strategies, as mentioned earlier the reason we can't use expectimax is because we have hidden information (the hole card of the opponents) , but what we can do is to model the opponent and act in the expectimax algorithm as the opponent model would act.
there are different way to model an opponent:

1. Neural Nets - (add reference) learn a model from historical games (individual to each player)

2. Bayesian model - (maybe will be the advance of the research) as we model

the rules of poker we could model our belief on opponent hidden information, and could maybe deduce some information given the actions the opponent did (posterior), the algorithm can combine this posterior distribution with the visible information distribution (hole cards and community cards) can try to optimize our moves given this distribution, could be from simulation driven approach or other approaches. this is an interesting study case because we combine the distribution of the stochastic nature of the game and the distribution of the hidden variables (opponents hole cards) , maybe something like (Seaman, 2018)

3. other ways to model the opponent - (TODO expand).

2.6 Case-based reasoning

The motivation to this approach is as the agent plays , it observes more and more cases of game states and the results that led by those state , the can reason on a new case , based on history cases, if it's a new case we may use K-nearest-neighbours algorithm on the history cases. some agents were models this way , one approach were that the agents starting with empty-case history and play random and add cases during the game. other approaches combine methods mentioned earlier with case-based reasoning , tend to be more case-based agent as more cases are observed.

2.7 Evolutionary algorithms and neural networks

2.8 Bayesian poker

2.9 Further Reading

1. TBB - theory-base Bayesian

2. POMDPs

3. Rapidly Exploring Random Tree

4. Marco F. Cusumano-Towner and Vikash K. Mansinghka. Encapsulating Models and Approximate Inference Programs in Probabilistic Modules. Computing Research Repository

5. Nick Chater, Joshua B Tenenbaum, and Alan Yuille. Probabilistic Models of Cognition: Conceptual Foundations.

6. Frith and Frith

7. Chris Frith and Uta Frith. Theory of mind

8. Effective Bayesian Inference for Stochastic Programs.

9. Vikash Mansinghka, Daniel Selsam, and Yura Perov. Venture: A Higher-Order Probabilistic Programming Platform with Programmable Inference T. C. Schelling. The Strategy Of Conflict. 1960

10. Andreas Stuhlm¨uller and Noah D Goodman. Reasoning About Reasoning by Nested Conditioning: Modeling Theory of Mind with Probabilistic Programs.. Probabilistic Inference for Solving (PO)MDPs

2.10 Experiment Settings

1. pokerwars - seems good but not ”scientific” - only web platform

2. pokergeniues - seems the most serious one - cost money

3. PyPokerEngine - opensource and free - looks for now like the best option

References

Moravčík, M., Schmid, M., Burch, N., Lisý, V., Morrill, D., Bard, N., . . . Bowling, M. (2017, Mar). Deepstack: Expert-level artificial intelligence in heads-up no-limit poker. *Science*, *356*(6337), 508–513. Retrieved from `http://dx.doi.org/10.1126/science.aam6960` doi: 10.1126/science.aam6960

Rubin, J., & Watson, I. (2011, 04). Computer poker: A review. *Artificial Intelligence*, *175*, 958-987. doi: 10.1016/j.artint.2010.12.005

Seaman, I. R. (2018). Probabilistic programming for theory of mind for autonomous decision making..