Summary of DeepStack: Expert-Level AI in Heads-Up No-Limit Poker

https://arxiv.org/pdf/1701.01724.pdf

Imperfect information games provide a significant challenge as compared to deterministic games. Since limited information is available to the agent, the agent most calculate a statistical best choice(s). Formally the statistical best response is accomplished by approximating the Nash equilibrium given the private information held by the opponent. DeepStack builds upon the traditional technique known as Counterfactual regret minimization, which is a recursive algorithm which would pre-compute the probability distributions. The issue with this previous method is that the computational requirements for Heads-Up No-Limit Poker (**HUNL**) approaches 10^{160} therefore approximates to game states would have to be made. A goal of DeepStack was to eliminate as much contextual information loss as possible.

When player a two-player zero sum game such as **HUNL** or goal is to maximize the expected utility against a best response strategy. Deep Stack calculates this through 3 elements. The first generates a representative picture of all currently available public information. Public information in the case would be the agents two cards in hand, the pot, and any visible cards dealt to the table. The second component is a depth limited search technique which uses a trained neural network to quickly evaluate values below a certain depth. By using the flexibility of neural networks to generate values, the depth needed to search can be greatly limited. Limited set of lookahead branches. By limiting to a known set of optimal actions, the recursive traversal computational cost can be reduced.

The internal representation of all public information is regenerated each action by a process the authors call “Continual re-solving”. This is done with two values, obviously, the first required values are the public state. Secondly, a vector of counterfactual values of valid “what if’s” relevant to the opponent. The counterfactual values start with a default starting state for the first round. Afterwards the vector is updated by output of the recalculation.

In order to make re-solving practical, the depth of the search is limited to four actions. Values are calculated through a neural network the authors call “**Deep Counterfactual Value Networks**”. Given the complexity created by imperfect information, each vector must contain what amounts to a description of the poker game being played. Concretely, a representation of the probability distribution of the cards dealt to the private hand and a value of the current knowns. The authors liken this output to “Intuition”. The network itself consists of seven hidden layers with 500 nodes each. Trained on 10 million randomly generated poker games for the turn and 1 million for the flop.

DeepStack has shown significant improvement over previous systems. Competing against various professional human players. Playing a total of 44,852 games. DeepStack won 492 mbb/g. Which the authors indicate as over 4 standard deviations away from zero. In comparison to previous algorithms such as Claudico(2015) which lost at a rate of 91 mbb/g. DeepStack shows a novel approach to relegating to neural networks strength in determining probability distribution over a wide range of outcomes. This system which demonstrates the possibilities in the future of AI in regards to solving problems with imperfect information.