Research Review: DeepStack: Expert-Level Artificial Intelligence in Heads-Up No-Limit Poker

In previous attempts for defeating poker players with computer programs, strategies were exploited and they were beatable with big margin. So, main target with DeepStack is beating human poker players in HUNL Poker with statistical significance by using different techniques than predecessors.

Term definitions related with game

- *Private information:* hands of two cards dealt face down
- *Public state*: consisting of the cards laying face up on the table and the sequence of betting actions made by the players.
- *Counterfactual value (CFV)*: conditional "what-if" value that gives the expected value if the opponent reaches the public state with a particular hand.
- *Decision point*: the combination of the public state and the hand for the acting player.
- (*Player's) strategy*: probability distribution over valid actions for each decision point.
- (*Player's) range value*: probability distribution over the player's possible hands given that the public state is reached.
- *mbb/g*: milli-big blinds per game

Techniques:

The DeepStack algorithm is composed of three ingredients:

1- A sound local strategy computation for the current public state: Continual resolving
Goal with continual resolving is to avoid maintaining a strategy for the entire game. Without having to solve entire game each step, reconstruction of strategy when need to act and never use it beyond next action. For that just needs to keep track of our own range and vector of opponent counterfactual values .These values are upper bound for current public state and lower than the deviated from reaching the public state.

In our turn, computed strategy is formed by using stored our range and opponent counterfactual values. Our action will be made by this new strategy, not prior ones. Our range values and opponent CFVs is updated by rules as defined in (i)own action and (ii)chance action. In (iii)opponent action, no change is required for both values.

Continual resolving never keeps track of opponent's range, instead only keeps track of opponent CFVs. It never requires knowledge opponent's action to update own range values and opponent CFVs. This is important difference from traditional resolving. So, it is efficient and no need translation step with action abstraction method. However, by itself continual resolving is impractical so depth and breadth is should be limited fro re-solved subtree.

2- Depth-limited lookahead using a learned value function
Due to poker is imperfection game and using iterative algorithm, players' range and counterfactual values change in each iteration. DeepStack solved this problem by creating replacing subtrees beyond a certain depth with a learned CFV function. Inputs of these function describes poker game and revealed cards. Outputs are estimation of how valuable holding certain cards in such a game.

Deep learning is used as learn function in DeepStack. Inputs are range of each players,pot size and public cards.The output from the seven fully connected hidden layers is vector for each players' CFVs of holding each hand. With depth limit is assigned as 4, decision points number is reduced from 10^{160} to 10^{17} before the resolving.

Depth-limited continual re-solving is sound, so intuation is “good” and “enough” computation is used in each re-solving step, then DeepStack plays an arbitrarily close approximation to a Nash equilibrium.

<u>3- A restricted set of lookahead actions: Sparse lookahead trees</u>
Construction sparse lookahead trees is reduction number of actions by defining only possible actions as fold,call,2 or 3 bet actions and all in. It helps to reducing decision points and playing at human speed.

<u>Results:</u>

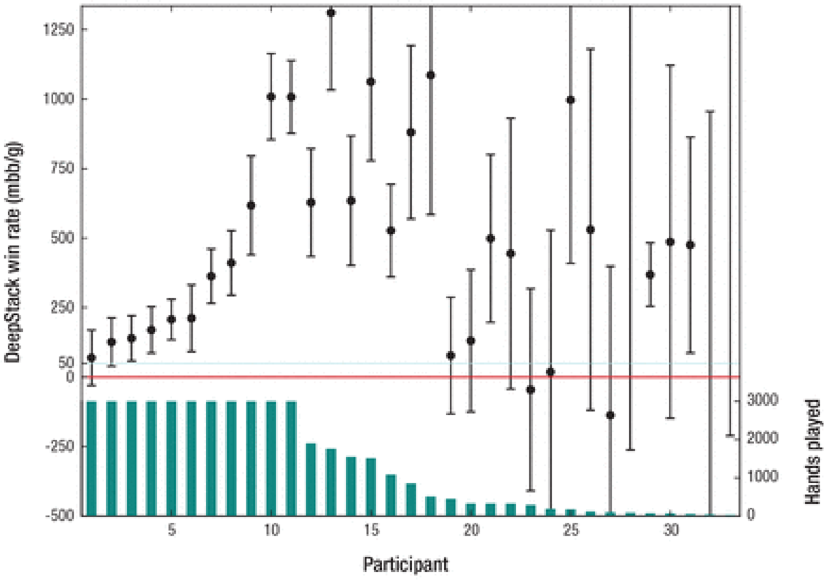

Figure 1: Performance of professional poker players against DeepStack.

In total 44,852 games were played by the 33 players, DeepStack won 492 mbb/g. This is over 4 standard deviations away from zero, highly significant.(50 mbb/g a sizable margin). Using AIVAT,a provably unbiased low-variance technique, to evaluate; DeepStack’s estimated performance actually 486 mbb/g. For this, margin is over 20 standard deviations from zero.