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

Lasse Becker-Czarnetzki

University of Heidelberg
Artificial Intelligence for Games
SS 2019

July 11. 2019
1. Perfect vs Imperfect information Games
   - Introduction
   - No-Limit Heads-up Texas Holdem
   - Perfect Information strategies

2. DeepStack
   - Re-solving (CFR)
   - Depth limited search
   - Counterfactual Value Networks
   - Sparse lookahead trees

3. Evaluation
   - Performance against humans
   - Exploitationability (LBR)
   - Nice features

4. Conclusion
Perfect information games
Von Neuman on games

Real life is not like that. Real life consists of bluffing, of little tactics of deception, of asking yourself what is the other man going to think I mean to do. And that is what games are about in my theory.

von Neumann from a discussion recounted by Bronkowski (1973)
No-Limit Heads-up Texas Holdem

- 2 player zero-sum game
- 4 Betting rounds on "who has the better cards"
- 2 Hold cards (private) (3, 4, 5) public cards.

$\rightarrow 10^{160}$ decisionpoints
Poker Terms

- Bigblind
- Fold
- Check
- Call
- Bet (raise)
- Flop (Pre-Flop)
- Turn
- River
- range
Poker Game Tree
Perfect information game
Perfect information game
Perfect information game
Perfect information game
Problems for imperfect information games
Questions

- How can we forget supergames without using necessary information?
- How do we solve a subgame when there are no definite states to start from?
- How do we evaluate a state, when we can’t use a single value to summarize a position?
Re-solving

DeepStack:
- re-solving
Re-solving

DeepStack:
- re-solving
Re-solving (CFR)
Counterfactual Regret Minimization

- **Counterfactual**: "If i had known"
- **Regret**: "how much better would i have done if i did something else instead?"
- **Minimization**: "what strategy minimizes my overall regret?"
- Average strategy over i iterations $= \text{approximation to Nash Equilibrium}$
Counterfactual Regret Minimization

Regrets = [0, -50, 50]

Reward = $50

Hypothetical Reward = $100

Hypothetical Reward = $0
Counterfactual Regret Minimization

Regrets = [0,0]

Reward = $50

Regrets = [0,−50,50]

Reward = $50

Hypothetical Reward = $100

Hypothetical Reward = $0
Counterfactual Regret Minimization

Regrets = [0, -550]

Regrets = [0, -50, 50]

Regrets = [0, 600]

Reward = $50

Hypothetical Reward = -$500

Reward = -$500

Reward = $50

Hypothetical Reward = $100

Hypothetical Reward = $0
Continual Re-solving

- At every action we re-solve the subgame
- We need our range and opponents counterfactual value ”What-if” (expected value) opponent reaches public state with hand x.
- 3 scenarios for updating range and CFVs.
  - **own action:** $CFVs = CFVs(\text{action})$ – Update range via Bayes rule
  - **Chance action:** $CFVs = CFVs(\text{chance action})$ – Eliminate impossible card combos.
  - **Opponents action:** Do Nothing
Depth limited search

DeepStack:
> depth-limited CFR
Depth limited search

\[ V(\text{range}) = \text{CFVs} \]
Solutions

- Search from a set of possible states, re-solving multiple times.
- Remember players range and opponents counterfactual values
- Get evaluation through Deep Counterfactual value networks
DeepStack elements summary
Deep Counterfactual Value Networks
Deep Counterfactual Value Networks

- 2 Networks: Flop Network, Turn Network
- Auxiliary network (Pre-Flop)
- Simple FFNN (7 layers, 500 Nodes, ReLU)
- outer network to fit values for zero-sum game
- **input:** Pot sizes, public cards, players ranges
- **output:** Counterfactual Values (Players, Hands)
Training

- Randomly generated Poker situations.
- Turn network: 10M, Flop network: 1M
- Turn network used for depth-limited lookahead in Flop Network training.
Sparse lookahead trees
Traditionally abstraction was used to simplify the game

Action abstraction – Card abstraction
   –> Translation Errors

Deepstack only uses action abstraction in lookahead

Card clustering is used for NN input.
Evaluation

- Exploitability – Play against humans
- Problems with Variance (Luck) → 100.000 Hands for statistical significance
  → AIVAT 3k Hands = 90k normal hands
Pro players experimental results
Pro players experimental results

<table>
<thead>
<tr>
<th>Player</th>
<th>Rank</th>
<th>Hands</th>
<th>Luck Adjusted Win Rate</th>
<th>Unadjusted Win Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Martin Sture</td>
<td>1</td>
<td>3000</td>
<td>70 ± 119</td>
<td>-515 ± 575</td>
</tr>
<tr>
<td>Stanislav Voloshin</td>
<td>2</td>
<td>3000</td>
<td>126 ± 103</td>
<td>-65 ± 648</td>
</tr>
<tr>
<td>Prakshat Shrimankar</td>
<td>3</td>
<td>3000</td>
<td>139 ± 97</td>
<td>174 ± 667</td>
</tr>
<tr>
<td>Ivan Shabalin</td>
<td>4</td>
<td>3000</td>
<td>170 ± 99</td>
<td>153 ± 633</td>
</tr>
<tr>
<td>Lucas Schaumann</td>
<td>5</td>
<td>3000</td>
<td>207 ± 87</td>
<td>160 ± 576</td>
</tr>
<tr>
<td>Phil Laak</td>
<td>6</td>
<td>3000</td>
<td>212 ± 143</td>
<td>774 ± 677</td>
</tr>
<tr>
<td>Kaishi Sun</td>
<td>7</td>
<td>3000</td>
<td>363 ± 116</td>
<td>5 ± 729</td>
</tr>
<tr>
<td>Dmitry Lesnoy</td>
<td>8</td>
<td>3000</td>
<td>411 ± 138</td>
<td>-87 ± 753</td>
</tr>
<tr>
<td>Antonio Parlavecchio</td>
<td>9</td>
<td>3000</td>
<td>618 ± 212</td>
<td>1096 ± 962</td>
</tr>
<tr>
<td>Muskan Sethi</td>
<td>10</td>
<td>3000</td>
<td>1009 ± 184</td>
<td>2144 ± 1019</td>
</tr>
</tbody>
</table>
Exploitability

- DeepStack '16
- AllCards '15 (100BB)
- Act1 '16
- Slumbot '16
- Hyperborean '14
- Always Fold

mbb/g
Nice to know

Thinking Time: 3s / action
7.2s / hand
Nice to know

Any Stack Size
Heads-up Freezeouts
DeepStack beats Pro Poker player in No-Limit Heads-Up Holdem for the first time

Connects Perfect information AI heuristical search strategy with imperfect information AI

Plays with Nash Equilibrium approximated strategy
→ Doesn’t exploit weaker players.

No Multiplayer

Can’t explain moves but strategy tips can be taken away from DeepStacks play.
References

DeepStack: Expert-Level Artificial Intelligence in No-Limit Poker
Science, 2017

Proceedings of the Twenty-Eighth Conference on Artificial Intelligence(2014)
pp. 602–608
References

Science 347
145 (2015)

[4] Todd W. Neller and Marc Lancto
An introduction to counterfactual regret minimization.
InProceedings of Model AI Assignments
The Fourth Symposium on Educational Advances in Artificial
Intelligence (EAAI-2013), 2013


[7] https://www.youtube.com/watch?v=qndXrHcV1sM
Thank You for Listening

Any Questions?