Limit Hold’em

Green Kiats

Papers

Bowling et al (2015) Heads-up limit hold'em poker is solved, *Science*, 9 January 2015, 347(6218): 145-149.

Risk and Szafron (2010) Using counterfactual regret minimization to create competitive multiplayer poker agents, *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems*, 1(1): 159-166.

Mazrooei, Archibald, and Bowling (2013) Automating collusion detection in sequential games, *Twenty-Seventh AAAI Conference on Artificial Intelligence Proceedings*.

Hold’em

- Game consists of multiple hands
- Winner of game has all chips in game
- Winner of hand determined by ranking of hand assembled
- Each player is dealt 2 private cards
- 5 cards are placed on the table
  - Flop, Turn, River

Hold’em

- Players have a stack of chips
- Chips in game is fixed number
- Betting occurs as cards are revealed
- Folding
- Players serve as the dealer
  - Bid last, which is commonly seen as an advantage

Limit Hold’em Poker?

- Limited betting
- Head’s up = 2 player
- Also studied 3 player

Why?

- Poker is a very complex game that people actually play
- Imperfect information
- Extensive-form game

Three Papers

- Two person limit hold’em
- Three person limit hold’em
- Collusion in three-person limit hold’em

Counterfactual Regret Minimization (CFR)

- Regret = utility lost for not selecting the best strategy, which can only be known at the end
- Looks for a solution, not all solutions
- Epsilon Nash equilibrium

CFR

- Alternative to normal-form linear programming
- Only store most recent regret, so linear memory requirements

CFR Variant (CFR+)

- Compress regrets using scaling
  - 292 TB reduced to 11 TB storage needed
- No negative regrets, so strategies retried earlier
- Since current strategy has an exploitability of near zero (empirically verified), CFR+ uses existing average regret instead of recomputing
  - Exploitability is the difference between money gained and expected worst-case performance

Computation of Results

- 200 Machines with 24 2.1-GHz AMD cores, 32GB of RAM, and a 1-TB local disk
- 68.5 days
- Divided subproblems to each machine using games up to the betting after the flop

Findings for this Paper

- Raising is preferred to calling for first player
  - “Limping” is a part of the optimal solution .06 percent of the time
- Dealer has an advantage
- Human players fold more than this equilibrium strategy

Limitations on Findings

- Assumes players are rational
- This is only one Nash equilibrium
- 1-in-20 chance of winning against its worst-case adversary in human lifetime of games

Concept of Solved

- Ultra-weakly solved
  - Game theoretic value computed
- Essentially weakly solved
  - ϵ-Nash equilibrium
- Weakly solved
  - Nash equilibrium computed
- Definitions neglect imperfect-information games

3-Player Limit Hold’em

Unsimplified number of game states:

- 2-player limit hold’em: 10^{18}
- 3-player limit hold’em: 10^{24}

1 “million billion billion” game states

Winning Strategies

- It is impossible to find ϵ-Nash Equilibria for a multiplayer game
- CFR has no guarantee for good performance in a multiplayer game

3-Player Games

Ante 1 unit
Choose heads or tails
All three same, antes returned
Two same, two win and split loser’s ante

Rule Simplications

3-Player Kuhn Poker

- 1 chip ante
- 1 private card
- 4-card deck (K>Q>J>T, no ties)
- 1 betting round, 1 chip bets
- If there is an outstanding bet, a player may fold or call

3-Player Leduc Hold'em

- 1 chip ante
- 1 private card
- 8-card deck (K>Q>J>T, 2 suits)
- 2 betting rounds, 2-bet cap
- 2 chip preflop bet
- 4 chip flop bet
- 1 community card dealt before the flop bet
- Pot is split on a tie
- Paired beats unpaired

Two Types of CFR Agents

- Perfect Recall: Remembers hand valuations for all four rounds
  - 2-bucket hand abstraction
    - $2^4 = 16$ bucket sequences
- Imperfect Recall: Only remembers the valuation of the current hand
  - 16-bucket hand abstraction
    - $16^1 = 16$ bucket sequences

Benchmarking

Poki: Heads-up and multiplayer agent that won a multiplayer limit event in 2008.

Chump agents: Agents defined by simple probability triples (fold, call, raise)

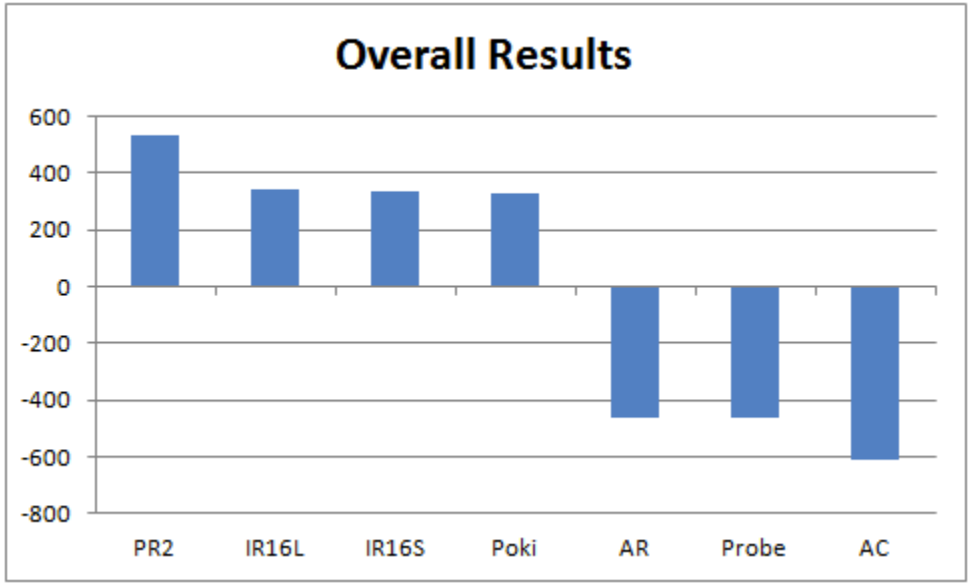
Chump Agents

Always-Fold (f,c,r) = (1,0,0) Folds
Always-Call (f,c,r) = (0,1,0) Checks or calls
Always-Raise (f,c,r) = (0,0,1) Bets or Raises
Probe (f,c,r) = (0,0.5,0.5) Equally checks/calls or bets/raises

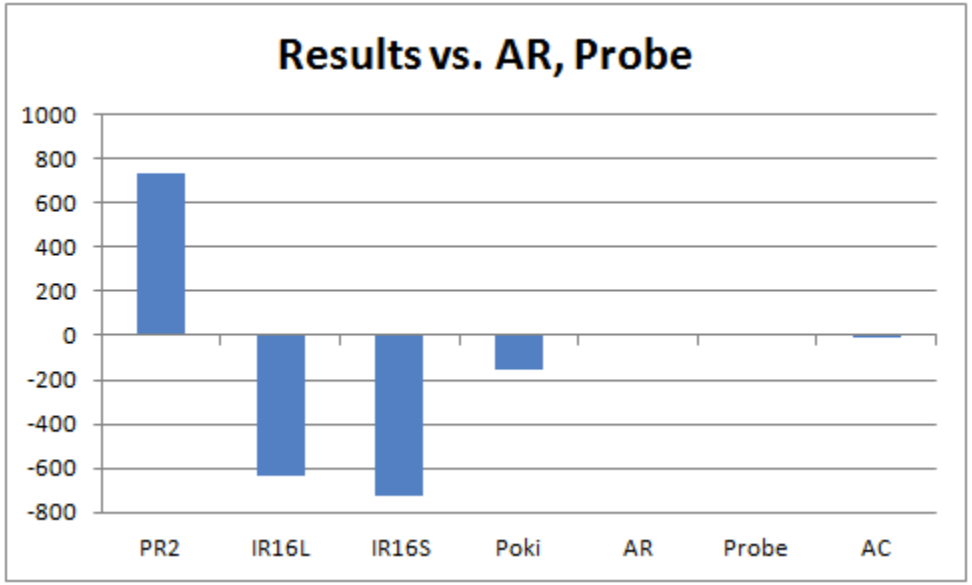
Bankroll and Elimination

Bankroll: Agents are ranked based on total win rate
Elimination: Single elimination where each round is ranked assuming the eliminated agent(s) had not entered until 3 agents remain.

Benchmarking Results



Benchmarking Results



2-Player Subgames

“Heads-Up Experts” are used when one player folds.

Require information about the pot and distribution of cards

Success in Competition

The 16-bucket imperfect recall agent placed first of 25 agents in the 2009 CP 3-player limit Hold'em competition

The 2-bucket perfect recall agent placed 2nd

Collusion

- collusion- 2 or more parties cooperating to detriment of others
- colluder plays as if his partner’s utility is valuable

3-player Limit [2-4] Hold’em

- 3-player limit Hold’em
  - 2 rounds only
  - 3 public cards
  - zero-sum extensive-form game with imperfect information
- CFR strategies abstracted to this form

Detecting Collusion

- assign collusion score to all pairs of agents based on behavior in many games
- higher scoring pairs should be investigated by humans for collusive activities

Collusion Value

- (C(j,k))- effect of agent k's actions on j's utility
- determined from value functions (Vi(h))
  - how much agent at position i might expect to win at end of game beginning at history h

Collusion Table

- captures effect of each player's actions on others' utilities in one game
  - table of C(j, k)s
    - effect of agent k's actions on j's utility
  - can incorporate chance "player"
    - episode utility uφg(j)(zg)

Collusion Table Example

Player k j	P1	P2	P3	Chance	Utility
P1	-3	+13	+2	-20	-9
P2	+8	-6	+2	-25	-21
P3	-5	-7	-3	+45	+30

Collusion Score

- designates degree of collusion exhibited by pair of agents in collusion table
  - Total Impact Score- sum of collusive values for pair
  - Marginal Impact Score- sum of difference in impact on partner - average impact on other agents for both agents

Testing Effectiveness

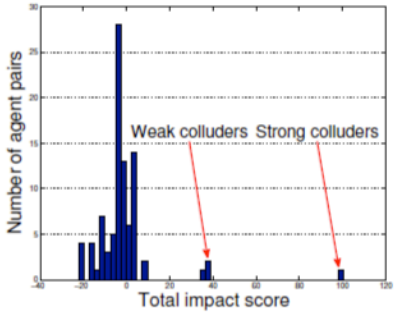
- created data set to test effectiveness
  - agent population
    - vary skill level (weak, strong)
    - vary positional CFR strategy (collusive, defensive, normal)
  - each possible 3-player configuration played 1-million hands (91 pairs)

Testing Effectiveness

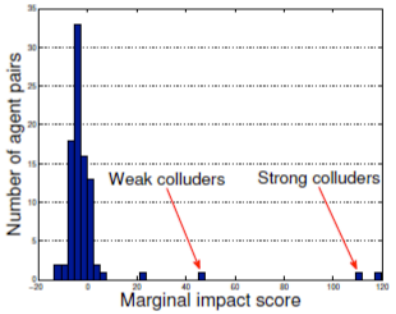
- two collusion detection methods with different value functions
  - Version A- determinized version of strong non-collusive strategy
  - Version B- determinized version of CFR strategy created via abstraction

Findings for Paper

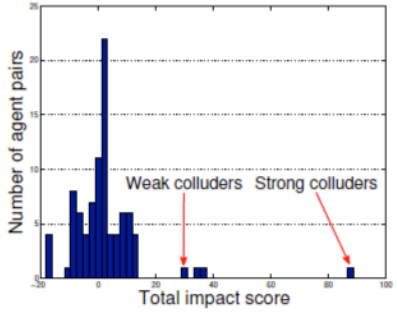
- weak and strong colluders are outliers
- collusion scores rank true colluders highly
  - A & B both detect colluding pairs => robust to choice of value function



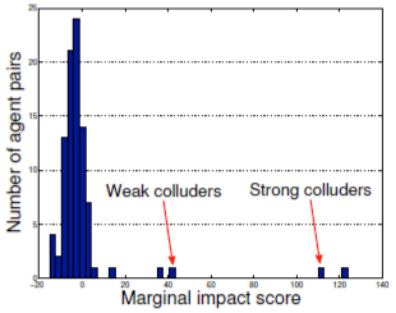
(a) Version A: TI Score



(b) Version A: MI Score



(c) Version B: TI Score



(d) Version B: MI Score

Findings for this Paper

- process detects accidental and intentional colluders
- process detects strong and weak colluders
- process is extensible to other zero-sum extensive form games with imperfect information

Applications

- Advises human players how to play more optimally
- A way of “solving” imperfect information, extensive form games
- Collusion detection

Critiques

- Strategies suggested don’t work well against all opponent types
- Not using data from actual games
- Some of the assumptions to make solutions more calculable were not well-justified