Efficient Nash Equilibrium Approximation through Monte Carlo Counterfactual Regret Minimization
Motivation

Tackling the practical challenge of Nash equilibrium computation in large games

♥ Strategy that is guaranteed to not lose on expectation (2-player, zero-sum)

♣ Very useful property in practice:
  ♦ Dominant approach in the Annual Computer Poker Competition
  ♠ 2008: beat human professionals at 2-player limit Texas hold’em poker
The poker community is now solving games with $10^9$ decisions (information sets).

LPs don’t scale to this size of game. We’ve made great progress on efficient approximation algorithms. (CFR, EGT)
Counterfactual Regret Minimization (CFR), NIPS 2007

CFR is the competition’s most popular algorithm.

Iterative, resembles self-play; reinforcement learning flavour.

- Memory efficient (2 doubles per infoset-action)
- Converges quickly ($1/\varepsilon^2$)
- Programmer Friendly
  - Easy to implement and optimize
  - Linear speedup with many cores

This paper: a new CFR variant that converges more quickly in imperfect information games.

Wednesday, November 14, 2012
Counterfactual Regret Minimization (CFR)

Basic idea:

♥ Start with two uniform random strategies. Play them against each other.

♣ Put a regret minimizing agent at every decision, and let it independently learn its part of the strategy.

♦ Run many iterations: walk the game tree, agents update their parts of the strategy.

♠ Average strategy profile converges to equilibrium.

\[ \text{Regret}(I) = (-2, 1, 4) \]
\[ \sigma(I) = (0, 0.2, 0.8) \]
Counterfactual Regret Minimization (CFR)

To update a decision, we need:
❤ Probability of other players taking their series of actions
♦ Expected value (or unbiased estimate) of actions’ utilities given opponent’s strategy

Recursively walk the tree:
♦ Push forwards opponent action probabilities
♠ Return EV at this terminal node or in this subtree

π_i(I) = 0.2
V(I) = (-2, 2, 6)
In practice, a sampling variant of CFR is used.

Chance Sampling: on each iteration, randomly sample one set of chance events and only update that part of the tree.

Terminal nodes: Get an unbiased estimate of my state’s value. Takes \(O(1)\) time.
New CFR Sampling Variants

Chance Sampling (CS)
Sample: Public chance
My Private Chance
Opponent Private Chance

Self-Public Chance Sampling (CS)
Sample: My Private Chance
Public chance
Expand: Opponent Private Chance

Opponent-Public Chance Sampling (OPCS)
Sample: Opponent Private Chance
Public chance
Expand: My Private Chance

Public Chance Sampling (PCS)
Sample: Public chance
Expand: My Private Chance
Opponent Private Chance

Wednesday, November 14, 2012
Opponent-Public Chance Sampling (OPCS)

公共机会

我的私人机会

对手私人机会

递归：
递出一个 scalar (对手到达概率)
返回一个 vector (子游戏的值)

关键观察：
对手无法观察到我的机会事件，所以他们的策略是相同的。
我可以有效地更新所有的这些决策！

终止节点：n 个状态要评估。

- 样本一个公共机会事件
- 样本一个对手的私人机会事件
- 列举所有可能的私人机会事件

心脏，星期三，11月14日，2012
New CFR Sampling Variants

Chance Sampling (CS)

Sample:
- Public chance
- My Private Chance
- Opponent Private Chance

Opponent-Public Chance Sampling (OPCS)

Sample:
- Opponent Private Chance
- Public chance

Expand:
- My Private Chance

Self-Public Chance Sampling (SPCS)

Sample:
- My Private Chance
- Public chance

Expand:
- Opponent Private Chance

Public Chance Sampling (PCS)

Sample:
- Public chance

Expand:
- My Private Chance
- Opponent Private Chance

Slower, Many updates per iteration

Wednesday, November 14, 2012
New CFR Sampling Variants

**Chance Sampling (CS)**
- Sample: Public chance, My Private Chance, Opponent Private Chance

**Opponent-Public Chance Sampling (OPCS)**
- Sample: Opponent Private Chance, Public chance
- Expand: My Private Chance

**Self-Public Chance Sampling (SPCS)**
- Sample: My Private Chance, Public chance
- Expand: Opponent Private Chance

**Public Chance Sampling (PCS)**
- Sample: Public chance
- Expand: My Private Chance, Opponent Private Chance

**Note:** Slower, Many updates per iteration

Wednesday, November 14, 2012
Self-Public Chance Sampling (SPCS)

Public Chance

Sample one Public chance event

My Private Chance

Sample one of my private chance events

Opponent Private Chance

Enumerate all of opponent’s possible private chance events

...(45 choose 2)

Recursion:
PASS one vector (opponent reach probabilities)
RETURN one scalar (value of subgame)

Heart: Much more precise estimate of my value, since I compare my state to all of theirs!

Club: Terminal nodes: \( n \) states to evaluate.

RESULT:
Slow but very precise updates.
New CFR Sampling Variants

**Chance Sampling (CS)**
- Sample: Public chance
  - My Private Chance
  - Opponent Private Chance
- Slower, many updates per iteration

**Opponent-Public Chance Sampling (OPCS)**
- Sample: Opponent Private Chance
  - Public chance
- Expand: My Private Chance

**Self-Public Chance Sampling (SPCS)**
- Sample: My Private Chance
  - Public chance
- Expand: Opponent Private Chance

**Public Chance Sampling (PCS)**
- Sample: Public chance
- Expand: My Private Chance
  - Opponent Private Chance

Wednesday, November 14, 2012
New CFR Sampling Variants

Chance Sampling (CS)

Sample:
- Public chance
- My Private Chance
- Opponent Private Chance

Expand:
- Opponent Private Chance

Opponent-Public Chance Sampling (OPCS)

Sample:
- Opponent Private Chance
- Public chance

Expand:
- My Private Chance

Self-Public Chance Sampling (SPCS)

Sample:
- My Private Chance
- Public chance

Expand:
- Opponent Private Chance

Public Chance Sampling (PCS)

Sample:
- Public chance

Expand:
- My Private Chance
- Opponent Private Chance

Slower, Many updates per iteration

Slower, Very precise updates

Wednesday, November 14, 2012
Public Chance Sampling (PCS)

- Sample one Public chance event
- Enumerate all of my private chance events
- Enumerate all of opponent’s possible private chance events
- Terminal nodes: $n$ states to evaluate against $n$ states. Looks like $O(n^2)$ work. But depending on game structure, $O(n)$ is often possible, making it as fast as OPCS or SPCS!

RESULT:
- Slower, but do many precise updates on each iteration.

Recursion:
- PASS one vector (opponent reach probability)
- RETURN one vector (value of subgame)
New CFR Sampling Variants

**Chance Sampling (CS)**
- Sample: Public chance, My Private Chance, Opponent Private Chance
- Slower, More updates per iteration

**Opponent-Public Chance Sampling (OPCS)**
- Sample: Opponent Private Chance, Public chance
- Expand: My Private Chance
- Same speed, very precise updates

**Self-Public Chance Sampling (SPCS)**
- Sample: My Private Chance, Public chance
- Expand: Opponent Private Chance
- Slower, very precise updates

**Public Chance Sampling (PCS)**
- Sample: Public chance
- Expand: My Private Chance, Opponent Private Chance
- Same speed, many updates per iteration
Results: 2-round, 4-bet Poker
94 million decision points (information sets)

Best response (mbb/g) vs. Time (seconds)
Abstracted Limit Texas Hold’em Poker

Real Poker Game

3*10^{14} Decisions (infosets)

Abstraction

Abstract Poker Game

10^9 Decisions (infosets)

Larger abstractions are better in practice, but take longer to solve.

Can evaluate by measuring exploitability in abstract game.
Results: Abstracted Limit Texas Hold’em Poker

- 5 buckets: 3.6m decisions
- 8 buckets: 23.6m decisions
- 10 buckets: 57.3m decisions
- 12 buckets: 118.6m decisions
Alternate domain:
Bluff, an imperfect information dice game
Conclusion

❤️ Counterfactual Regret Minimization is a state-of-the-art algorithm for Nash equilibrium approximation in 2-player zero sum games.

♣️ Public Chance Sampled CFR:
   ♦️ Takes advantage of structure of imperfect information
   ♠️ Converges faster in practice
Thanks!
Poster: Panel 056

Ground Floor

University of Alberta
Computer Poker Research Group

Wednesday, November 14, 2012
Games Solved for the Annual Computer Poker Competition

Size of Game Solved

- Number of Information Sets:
  - 2006: $10^5$
  - 2007: $10^6$
  - 2008: $10^7$
  - 2009: $10^8$
  - 2010: $10^9$
  - 2011: $10^{10}$

Distance to Equilibrium

- Exploitability (milliblinds/game):
  - 2007: 500
  - 2008: 400
  - 2009: 300
  - 2010: 200
  - 2011: 100

Competition Year Range: 2006 to 2011
Results: Limit Texas Hold’em Poker
One-on-One performance against a strong opponent (Hyperborean2010.IRO)