Finding Optimal Abstract Strategies in Extensive-Form Games

Mike Johanson, Nolan Bard, Neil Burch, Michael Bowling
University of Alberta, Canada
July 25th, 2012 :: AAAI 2012, Toronto
2-Player Limit Texas Hold’em Poker: Distance from Perfect Play

Exploitability (mbb/g)

Year

2006 2007 2008 2009 2010 2011

AAA! 2007
Vancouver: Narrow loss to Human Pros (275)

Las Vegas: Narrow win over Human Pros (235)

104

Tuesday, November 13, 2012
Abstraction-Solving-Translation

Goal:
We want to learn a strategy $\sigma$ (or, in RL, a policy $\pi$) that chooses actions.

Exploitability:
Expected loss against a perfect adversary.

Nash Equilibrium:
Unexploitable - expected loss of $0$ per game. An **optimal strategy**. We want to approximate this.
Problem:
The game has $10^{14}$ information sets. Far too large to solve!

With current techniques, this would take 4 petabytes of RAM and thousands of CPU-years!
Problem:
The game has $10^{14}$ information sets. Far too large to solve!

With current techniques, this would take 4 petabytes of RAM and thousands of CPU-years!

If you have four petabytes of RAM, we should talk!
Abstraction-Solving-Translation

Workaround: Use state-space abstraction to make a smaller game that we can solve.

Game $10^{14}$ decisions

Abstraction

Abstract Game $10^7$ decisions

Solver

Optimal Strategy
**Solving:**
Use a game-solving algorithm to find an optimal strategy for the abstract game.

---

**Abstraction:**
Use game-solving algorithm to find an optimal strategy for the abstract game.

**Game:**
10^{14} decisions

**Abstract Game:**
10^7 decisions

---

**Optimal Strategy:**

**Optimal Abstract Strategy:**
**Abstraction-Solving-Translation**

- **Game** $10^{14}$ decisions
  - Abstraction
  - **Solver**
  - **Strategy**

- **Abstract Game** $10^7$ decisions
  - **Solver**
  - **Optimal Abstract Strategy**
  - **Translation**

**Solving:**
Use a game-solving algorithm to find an optimal strategy for the abstract game.
Abstraction-Solving-Translation

Game $10^{14}$ decisions \rightarrow\text{Abstraction} \rightarrow \text{Abstract Game $10^7$ decisions}

- **Solver**
  - **Optimal Strategy**
  - $\not=\not=\not=\not=\not=$
  - **Optimal Abstract Strategy**

**Two Types of Loss:**

- **Lossy abstraction.** May not be possible to represent an optimal strategy.
- Other abstract strategies might be better in the real game!
Abstract Equilibrium might not be optimal in the real game.
Abstract Equilibrium might not be optimal in the real game.
This Talk:
Efficiently finding an abstract strategy with the lowest exploitability in the real game.
Counterfactual Regret Minimization (CFR)  
NIPS 2007

\[ \sigma^0 = \text{uniform random} \]

\[ t=0 \]
“Play a game”,
Update using CFR

$\sigma^0$

$\sigma^1$

vs

Updating with CFR makes them regret-minimizing agents.

NIPS 2007
Counterfactual Regret Minimization (CFR)  
NIPS 2007

The “Current” strategy

\[ \sigma^0 + \sigma^1 + \sigma^2 \]

The “Average” strategy

\[ \frac{\sigma^0 + \sigma^1 + \ldots + \sigma^t}{t} \]
If both players are regret-minimizing, then their average strategy converges towards an optimal strategy.
Counterfactual Regret Minimization (CFR)  
NIPS 2007

CFR in an abstract  
10-Bucket Perfect Recall Game

Abstract Game  
Exploitability

CFR Iterations

Abstract Game

Exploitability
Counterfactual Regret Minimization (CFR)  
NIPS 2007

CFR in an abstract  
10-Bucket Perfect Recall Game

Abstract Game  
Exploitability

Real Game  
Exploitability

CFR Iterations

Abstract Game  
Real Game

Tuesday, November 13, 2012
Moving from CFR to CFR-BR in six easy steps.
Both players abstracted.

Computation is efficient,
Solution is suboptimal.
$X$ is typically 1 to 100, depending on size of abstraction.

Both players are abstracted.

$X$ GB RAM

$X$ GB RAM
Opponent is unabstracted.

[Waugh et al., 2009]:
Opponent is unabstracted.
Abstracted player minimizes exploitability!
Requires far too much RAM and computation.
Play against a Best Response.

Abstracted, CFR

VS

Unababstracted, Best Response

100 GB RAM

8.75 TB RAM

A Best Response is also regret-minimizing, so average CFR strategy converges. Current CFR strategy converges, too! Takes 76 CPU-days to compute a BR.
Split Best Response into pieces.

Split strategy into a **Trunk** and many **Subgames**.

Big advantage of Best Response:
Can compute subgames independently as needed!
Never need to store all of it at once!
4. Split Best Response into pieces.

Abstracted, CFR vs Compute subgames as needed, then discard. Memory problem solved! Takes 2x76 CPU-days, though: first pass to compute Trunk, second pass to play the game.
Play against a **CFR-BR** Hybrid.

Use CFR to update Trunk strategy. This is also **regret-minimizing**, so CFR converges. Can query Trunk strategy any time, and compute Subgame strategy as needed.
Use Sampling to converge faster.

Sample one subgame, compute BR, update players. Takes **50 CPU-seconds** per iteration and **940 MB RAM**, and still converges!
**CFR-BR:**
Finds the least exploitable abstract strategy, while using *less* RAM than CFR did!

Average Strategy: Guaranteed to converge.  
Current Strategy: Not guaranteed, but converges faster in practice.
Testing in a small poker game


<table>
<thead>
<tr>
<th>Time (CPU-seconds)</th>
<th>Exploitability (mbb/g)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>10^3</td>
</tr>
<tr>
<td>10^-1</td>
<td>10^2</td>
</tr>
<tr>
<td>10^-2</td>
<td>10^1</td>
</tr>
<tr>
<td>10^-3</td>
<td>10^0</td>
</tr>
<tr>
<td>10^-4</td>
<td>10^-1</td>
</tr>
<tr>
<td>10^-5</td>
<td>10^-2</td>
</tr>
<tr>
<td>10^-6</td>
<td>10^-3</td>
</tr>
<tr>
<td>10^-7</td>
<td>10^-4</td>
</tr>
</tbody>
</table>

CFR
CFR-BR Average
CFR-BR Current

Tuesday, November 13, 2012
Testing in a small poker game

Abstracted [2-4] Hold’em:
1790 information sets
Texas Hold’em Poker: Small Abstractions

2007 Computer Poker Competition Abstraction

57 million information sets

(Previous best strategy: 100x larger abstraction, exploitable for 104)

Exploitability (mbb/g)

Time (CPU-seconds)
Texas Hold’em Poker: Tiny Abstractions

2-Bucket and 3-Bucket Abstractions:
These fit on a 1.44 MB Floppy Disk!

(2008 Man-vs-Machine Winner: 1.25 GB, exploitable for 235)

Tuesday, November 13, 2012
Texas Hold’em Poker: Small Abstractions

Least Exploitable Strategy Ever Made:
5.8 Billion information sets

Hyperbolean 2011.IRO
CFR-BR Average
Previous Best Strategy, Same Abstraction: 104

Time (CPU-seconds)

Exploitability (mbb/g)
2-Player Limit Texas Hold’em Poker: Distance from Perfect Play

Exploitability (mbb/g)

Year

2006 2007 2008 2009 2010 2011 2012

Narrow loss to Human Pros

Narrow win over Human Pros

This Talk: CFR-BR

Tuesday, November 13, 2012
Thanks!
More results at the poster!

Exploitability (mbb/g)

Year

2006 2007 2008 2009 2010 2011 2012

Narrow loss to Human Pros
Narrow win over Human Pros
This Talk: CFR-BR
Bonus Slides
One-on-One: PR 10s

#### Time (CPU-Seconds)

- **Time (CPU-Seconds)**
  - 10^5
  - 10^6
  - 10^7
  - 10^8

#### One-on-One (mbb/g)

- **CFR-BR Current**
  - -23.9
  - -26.2

- **CFR-BR Average**

- **One-on-One (mbb/g)**
  - -150
  - -125
  - -100
  - -75
  - -50
  - -25
  - 0
  - 25
  - 50
  - 75
  - 100
  - 125
  - 150

Tuesday, November 13, 2012
One-on-One: IR 9000

![Graph showing time vs. one-on-one (mbb/g)]

- CFR-BR Current
- CFR-BR Average

Tuesday, November 13, 2012
One-on-One: vs 2011

-7.5
-18.2
-27.9
-39.9

Time (CPU-Seconds) vs One-on-One (mbb/g)