Accelerating Best Response Calculation in Large Extensive Games

Michael Johanson, Michael Bowling - University of Alberta
Kevin Waugh - Carnegie Mellon University
Martin Zinkevich - Yahoo! Inc

How strong are recent computer poker programs?

A popular approach to making computer poker agents: approximate a Nash Equilibrium strategy. This minimizes the worst-case loss.

- Ties against a worst-case opponent (a best response), and may win against weak opponents

**My view of the game**

1400 cpu years
4 PB RAM

**Opponent's view of the game**

1 day (24 cpu) +1 gig RAM

Expectimax Search

To determine our payoff at A, we need to know the distribution over the opponent being in X and Y.

Recursive tree walk algorithm: PASS FORWARD.

An array of probabilities of the opponent being in each of their information sets (X and Y).

Our value at our information set, given the opponent distribution.

Only visits each game state once. But in big domains (10^4 in our game) this is intractable.

4 steps for accelerating best response computation in imperfect information games

1: Walking the Public Tree

2: Fast Terminal Node Evaluation

3: Avoid isomorphic states

4: Parallelize the computation

We don’t know if the opponent is at X or Y, but they don’t know if we are at A or B.

Walk the much smaller ‘Public Tree’:

- Far fewer nodes than our tree
- More work per node

11x speedup in practice.

New algorithm: PASS: Vector of reach probabilities, for their private states

RETURN: Vector of values, for our private states

Fundamental metric for game theory:

Man-vs-Machine matches

- Same approach as TAC, RoboCup, General Game Playing

Accelerating Best Response Calculation in Large Extensive Games

Four steps for accelerating best response

1. Walking the Public Tree

2. Fast Terminal Node Evaluation

3. Avoid isomorphic states

4. Parallelize the computation

We don’t know if the opponent is at X or Y, but they don’t know if we are at A or B.

Walk the much smaller ‘Public Tree’:

- Far fewer nodes than our tree
- More work per node

11x speedup in practice.

New algorithm: PASS: Vector of reach probabilities, for their private states

RETURN: Vector of values, for our private states

Trivial opponents

- Perfect Play
- Always-Caught

Best Response

Always-Fold 750
Always-Call 1163.48
Always-Raise 3697.69
Uniform Random 3466.32

A human pro aims to win at least 50 bb/g.

Abstraction Size (if information sets)

**Results in 2-player Limit Texas Hold’em**

**Trivial opponents**

<table>
<thead>
<tr>
<th>Name</th>
<th>Best Response (Millions per game)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Always-Fold</td>
<td>750</td>
</tr>
<tr>
<td>Always-Call</td>
<td>1163.48</td>
</tr>
<tr>
<td>Always-Raise</td>
<td>3697.69</td>
</tr>
<tr>
<td>Uniform Random</td>
<td>3466.32</td>
</tr>
</tbody>
</table>

**Best Response per time**

- Always-Fold
- Always-Call
- Always-Raise
- Uniform Random

**Tilting with Polaris 2008**

<table>
<thead>
<tr>
<th>Name</th>
<th>Payoff Adjustment</th>
<th>Best Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>White</td>
<td>0.036</td>
<td>263.702</td>
</tr>
<tr>
<td>Black</td>
<td>-0.036</td>
<td>257.231</td>
</tr>
<tr>
<td>Red</td>
<td>0.036</td>
<td>277.457</td>
</tr>
<tr>
<td>Green</td>
<td>-0.036</td>
<td>235.006</td>
</tr>
<tr>
<td>Hoover</td>
<td>0.036</td>
<td>235.006</td>
</tr>
</tbody>
</table>

**Best Response**

- Always-Fold
- Always-Call
- Always-Raise
- Uniform Random

**2010 Computer Poker Competition**

<table>
<thead>
<tr>
<th>Name</th>
<th>Payoff Adjustment</th>
<th>Best Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>White</td>
<td>0.036</td>
<td>263.702</td>
</tr>
<tr>
<td>Black</td>
<td>-0.036</td>
<td>257.231</td>
</tr>
<tr>
<td>Red</td>
<td>0.036</td>
<td>277.457</td>
</tr>
<tr>
<td>Green</td>
<td>-0.036</td>
<td>235.006</td>
</tr>
<tr>
<td>Hoover</td>
<td>0.036</td>
<td>235.006</td>
</tr>
</tbody>
</table>