Modelling an Opponent in Board Games

Julian Jocque
Motivation

- What if we could create a program to play exactly like Garry Kasparov?
Approach

- Use the Estimation-Exploration Algorithm to model opponents in Konane by presenting the opponent with board states.
Board State and Game Trees

http://www.ocf.berkeley.edu/~yosenl/extras/alphabeta/alphabeta.jpg

http://1.bp.blogspot.com/-1VruAl_cdE0/TwccPUduVTI/AAAAAAAACtg/6hlBEwxXeLI/s400/zz+larsen+petrosian+game+chessboard+r7_pp2pB2_3p3k_8_2PR4_8_PP4PP_5K2.gif
Minimax

http://s175.photobucket.com/user/habsq/media/minimax-2.jpg.html
## Static Evaluators

<table>
<thead>
<tr>
<th>I have a queen</th>
<th>They have a queen</th>
<th>Number of piece I have</th>
<th>Number of pieces they have</th>
<th>I am in checkmate</th>
<th>They are in checkmate</th>
</tr>
</thead>
<tbody>
<tr>
<td>+2.735</td>
<td>-1.4</td>
<td>+3.55</td>
<td>-2.78</td>
<td>-10000</td>
<td>+10000</td>
</tr>
</tbody>
</table>

Allows for Minimax to stop at a particular depth
Evolution

1. Generate Population
2. Evaluate Individuals
3. Select Best Designs
4. Evolve to Generate New Population

Used with permission from Ben Berger
Estimation Exploration Algorithm

1

3.78  2.225  -1.24  1.33  4.3  2.9
...   
-2.4  -3.6  1.45  1.11  2.0  1.8

2

3

4

2.41  9.978  1.43  -2.3  3.0  -1.2
...   
6.89  -1.13  -2.45  -4.1  9.1  -4.2
The System I Built

- Konane engine with Minimax, Alpha Beta Pruning
- EEA with evolving static evaluators and evolving sets of board states
- Model evaluator script
- All programmed in Python, all from scratch except Konane starter code
Running The System
Results

Fitness for each generation
Results cont.

• Data still needs interpretation
• Up to 90% accurate on opponents similar to models, only up to 65% on different settings
• About 45% accurate against opponent found on Github
Questions?