Using Reinforcement Learning to Play Othello

Kevin Fry (kfry)[1]; Frank Zheng (fzheng)[1]; Xianming Li (xmli)[1]
[1]Department of Computer Science, Stanford University; CS 229 Machine Learning

Stanford | ENGINEERING
Computer Science

Abstract

We built an AI that learned to play Othello. We used value iteration on a value approximation function in combination with minimax tree search with alpha-beta pruning. The performance of our four-feature linear value approximation function stands out above all other strategies in terms of win percentage.

Data & Methodology

Board Representation

- We represent the 8x8 board as a two-dimensional array, with each entry as a value from the set {-1,0,1}, which corresponds to {black, empty, and white}, respectively.

Gameplay

- AI used a shallow k-depth minimax tree search to play.
- The value of a node was approximated by our linear value function based on board features:

1. Frontier Disks: number of pieces adjacent to open spaces.
2. Mobility: number of available moves.
3. Piece Difference: proportion of the pieces on the board that are yours.
4. Value Matrix: score based on relative value of positions taken.

Algorithm

Tuning Value Function

- In tuning our value function, our agent trained against an opponent using a fixed set of weights according to the following algorithm:

Algorithm:

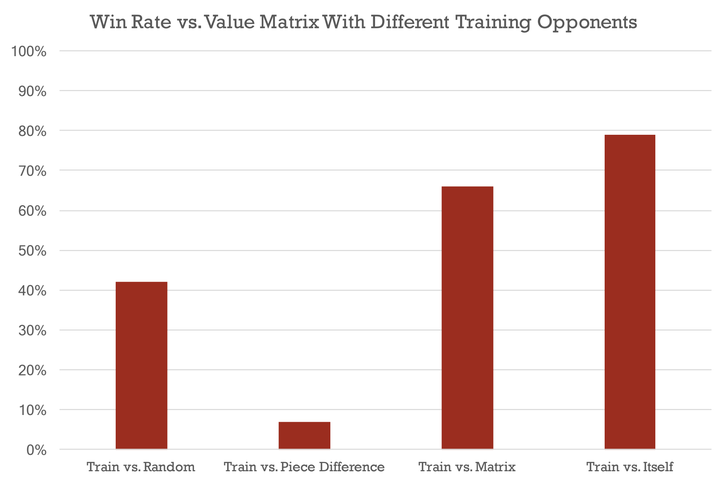
1. **repeat until convergence:**
2. Randomly initialize weights θ.
3. Every ten iterations update training opponent $\theta_{opp} = \theta$ (First ten iterations, opponent uses value matrix)
4. Based on exploration rate ϵ, either make the optimal move using current value function, or make a random move
5. For each move, record the board value v using current weights and feature vector f, and the corresponding minimax score v'
6. $\theta := \alpha(v' - v)f$

State Space Exploration

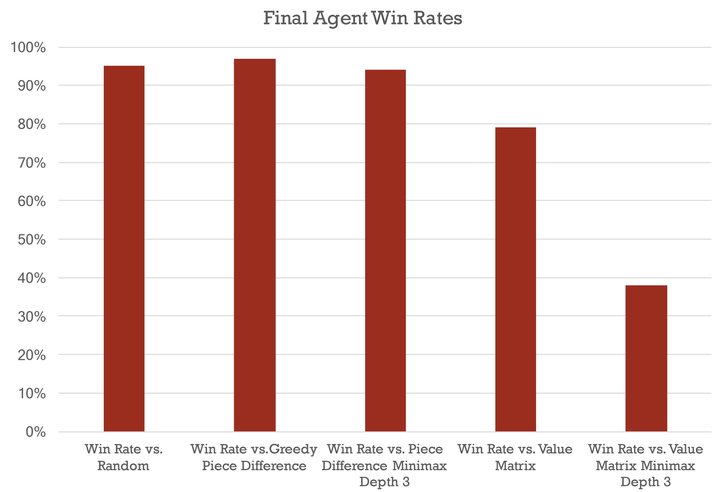
- Ensured state space is explored sufficiently to find optimal solution.
- Implemented an epsilon exploration rate of 0.5, decaying with each iteration.

Results

Comparing Training Opponents:

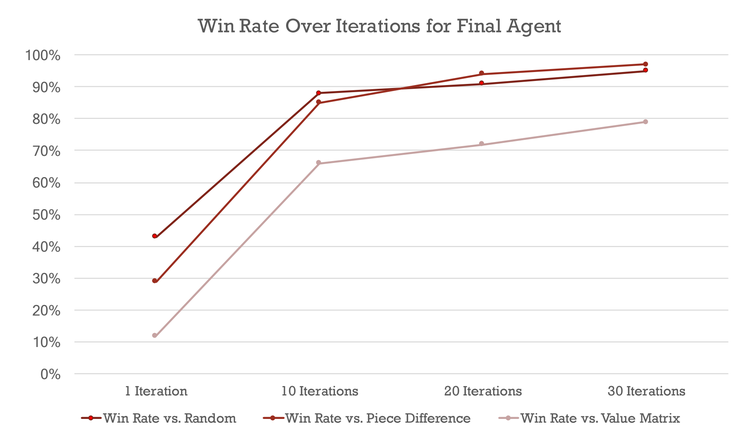


Win Rates for Final Agent:



Results

Final Agent Learning Curve (until convergence):



Discussion & Future Extensions

Our four-heuristic model outperformed benchmarks of random agent, piece difference (with and without minimax), and Korman matrix[1] with no minimax.

It failed to beat Korman matrix with minimax depth-3 even with considerably long training period, probably because Korman matrix was determined experimentally with prior biases, while ours was determined purely via unbiased learning.

There were several limitations to our agent:

- Our feature-set may be sub-optimal, future work could explore other features, and use deep learning or a feature selection algorithm to determine the optimal combination of features.
- Our current linear value function approximation likely does not model the value function well. A future version could use a neural network to better approximate the value function.
- In real-gameplay, there would be time-constraints on making a decision, so a future version of this agent should implement iterative deepening to ensure a solution is ready when time runs out.

References

1. M. Korman, "Playing Othello with Artificial Intelligence," 2003. [Online]. Available: http://mkorman.org/othello.pdf. Accessed: Nov. 11, 2016.
2. J. van Eck and M. van Wezel, "ScienceDirect," in ScienceDirect, 2006. [Online]. Available: https://pdfs.semanticscholar.org/eeec/5e39fdd0c8be0e6270c46488ccfe0254a8b6.pdf. Accessed: Nov. 11, 2016.
3. V. Sannidhanam and M. Annamalai, "University of Washington," in University of Washington, 2015. [Online]. Available: https://courses.cs.washington.edu/courses/cse573/04au/Project/mini1/RUSSIA/Final_Paper.pdf. Accessed: Nov. 11, 2016.
4. J. Festa and S. Davino, "'IAgo Vs Othello': An artificial intelligence agent playing Reversi,"2013. [Online]. Available: http://ceur-ws.org/Vol-1107/paper2.pdf. Accessed: Nov. 12, 2016.