Heuristic Analysis

Artificial Intelligence Nanodegree Program

Assignment 2 - Build a Game-playing Agent

Author: Alexander Ponamarev

Date: June 8, 2017

Governing questions 1 - Describe implemented functions and verbally describe the performance of agents using the implemented evaluation functions.

improved_score - provides a good estimate for the utility of the state. Improved score relies on the difference between remaining legitimate steps for the player and the opponent.

- custom_score (dubbed CenterToDelta Score) - combines the improved score heuristic and a center score. The center score measures the distance of a given player from the center of the game board. The goal of the custom score is to penalize player for deviating from the center of the board in the beginning of the game and rewards for advantage in available steps on the later stage. The temporal difference achieved through a mix of center score and improved (delta) score with weight exponentially annealing towards improved score.

- custom_score_2 (dubbed AggressiveCenterToDelta Score) - combines the improved score heuristic and a center score of the opponent. The goal of this function is to promote more aggressive player behavior and push the opponent away from the center of the game board. Custom_score_2 rewards a player for the distance of the opponent away from the center of the board and rewards the player for the advantage in available steps. This score is a mix of the opponent center score and the improved (delta) score with weight exponentially annealing towards improved score.

- custom_score_3 - combines both approaches (AggressiveCenterToDelta and CenterToDelta scores). This function combines rewards for pushing opponent away from the center and punishes player for deviating from the center of the board. In addition, this function rewards a player for advantage in available steps. This score is a mix of opponent and player center scores and improved (delta) score with weight exponentially annealing towards improved score.

Governing questions 2 - The report makes a recommendation about which evaluation function should be used and justifies the recommendation with at least three reasons supported by the data.

Based on the results of tournament.py presented below, custom_score function should be used for minimax agent with alpha-beta pruning for the following reasons:

1. Custom score outperforms the improved score vs. Minimax agent with open value function and improved value function, while performing on par for alpha-beta agent.
2. Overall, custom score function outperformed all competing functions with an average win rate of 73.1% over 30 runs (please see appendix for further detail). Improved function, is the second best choice with 70.2% win rate.
3. The difference in performance is statistically significant. The performance of value function may vary substantially depending on the initial position of the players. In order to eliminate initialization bias, the performance of value functions was tested on 30 runs (please see appendix for further detail). T-Test analysis confirmed statistical significance of the performance difference between custom score and improved score (Null hypothesis was rejected at P value=1.8% and t value = 5%).

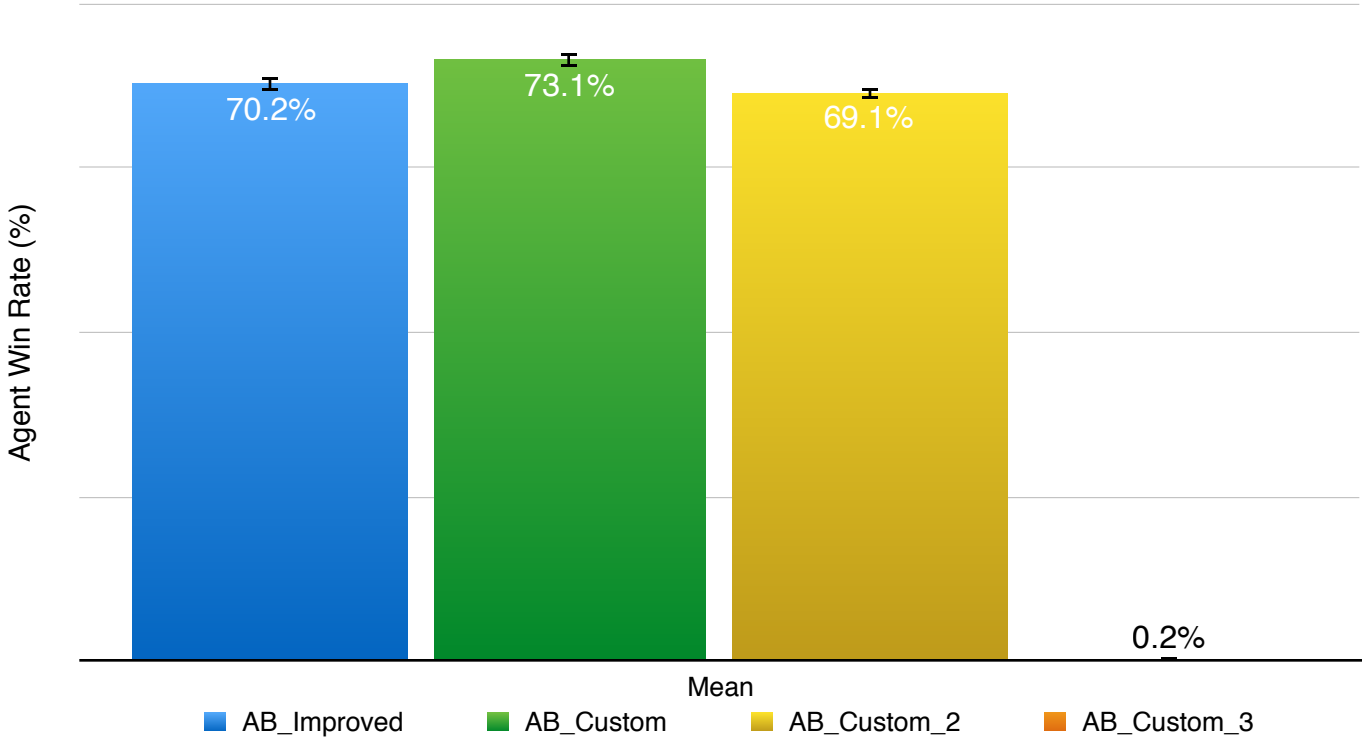
Appendix

Evaluation of value function:

Value function was tested based on script provided in tournament.py. The script was ran for 20 iterations with the following hardware configuration:

- AWS EC2 instances: fleet of 5 stop-request instances
- Instance type: c3.large
- AMI ID: udacity-aind2-0.0.6 (ami-2d6a1e3b)
- Conda: aind2

Result summary of 30 runs of tournament.py presented below:



Run Summary

	AB_Improved	AB_Custom	AB_Custom_2	AB_Custom_3
run 1	72.9%	70.0%	67.1%	1.4%
run 2	72.9%	77.1%	65.7%	0.0%
run 3	67.1%	67.1%	68.6%	0.0%
run 4	68.6%	71.4%	74.3%	0.0%
run 5	70.0%	75.7%	75.7%	0.0%
run 6	74.3%	74.3%	68.6%	0.0%
run 7	62.9%	74.3%	72.9%	1.4%
run 8	71.4%	74.3%	70.0%	0.0%
run 9	72.9%	62.9%	65.7%	0.0%
run 10	74.3%	62.9%	72.9%	0.0%
run 11	71.4%	77.1%	67.1%	0.0%
run 12	67.1%	71.4%	67.1%	0.0%
run 13	77.1%	80.0%	68.6%	0.0%
run 14	68.6%	75.7%	65.7%	0.0%
run 15	65.7%	71.4%	67.1%	1.4%
run 16	70.0%	80.0%	64.3%	0.0%
run 17	68.6%	81.4%	75.7%	0.0%
run 18	71.4%	68.6%	74.3%	0.0%
run 19	68.6%	68.6%	60.0%	0.0%
run 20	72.9%	67.1%	68.6%	0.0%
run 21	62.9%	74.3%	72.9%	1.4%
run 22	71.4%	74.3%	70.0%	0.0%
run 23	72.9%	62.9%	65.7%	0.0%
run 24	70.0%	75.7%	75.7%	0.0%
run 25	74.3%	77.1%	67.1%	0.0%
run 26	67.1%	71.4%	67.1%	0.0%
run 27	77.1%	80.0%	68.6%	0.0%
run 28	65.7%	71.4%	67.1%	1.4%
run 29	62.9%	75.7%	74.3%	0.0%
run 30	74.3%	80.0%	64.3%	0.0%
Mean	**70.2%**	**73.1%**	**69.1%**	**0.2%**
St. Error	**0.7%**	**1.0%**	**0.7%**	**0.1%**
P Value of Null Hypothesis:		**1.8%**	**26.4%**	**0.0%**