Heuristic Analysis

Dovydas Čeilutka

November 26, 2017

Three custom heuristic functions were created to evaluate the position value of the move in Isolation game. The performance of these functions were determined using three rounds of simulations using `tournament.py` script.

1 Custom heuristic functions

In the project three custom heuristic functions were designed and implemented:

1. center bias heuristic function

2. edge bias heuristic function

3. agressive attack heuristic function

1.1 Center bias heuristic function

Resoning The idea behind this heuristic function is that the closer the move is to the center the more valuable it should be. If a player after a move ends up near the edge or even worse in the corner this move should be penalized. On the other hand the closer the move brings the player to the center of the board the better future prospects.

Implementation The function is implemented by defining a helper function `center_value` function, which calculates the centeredness of the move using this function.

$$centeredness = \frac{1}{1 + |\frac{board_h}{2} - move_y - 0.5|} + \frac{1}{1 + |\frac{board_w}{2} - move_x - 0.5|}$$

where $board_h$ is the height of the board, $board_w$ is the width of the board, $move_y$ is the y position of the move and $move_y$ is the x position of the move.

Each available move is mapped throught the `center_value` function and the results are calculated for both the player and the opponent. The result is the difference between the player's and the opponent's values.

1.2 Edge bias heuristic function

Resoning This heuristic function has a similar idea to the center bias heuristic function, but it only cares about the edges ob the board. The idea is that the edges are especially bad and should be avoided, therefore all moves leading to the edge of the board are not counted.

Implementation The function gets all the available moves for the player and the opponent and filters out the moves, which direct to the edge of the board. The result of the function is the difference between the number of filtered moves for the player and the opponent.

1.3 Agressive attack heuristic function

Resoning This simple heuristic function calculates be difference between the moves available to the player and the moves available to the opponent, but the opponent's moves have a weight of 2. This will make the function very highly value the moves, which prevent the opponent from moving. This function was chosen, because of it's simplicity and speed.

Implementation The function gets all the available moves for the player and the opponent and returns the difference between the number of the player's moves and the number of the opponent's moves multiplied by 2.

1.4 Reinforcement learning of policy networks

In the next stage the policy network is improved using policy gradient reinforcement learning. The reinforcement learning policy network has identical structure to the supervised learning policy network. The weights are also initialized using the values from the previous step. Then games between random policy from previous iteration and current policy are played to prevent overfitting to the current policy and weights are updated.

1.5 Reinforcement learning of value networks

In the final stage the value function is estimated for position evaluation. Optimal value function under perfect play is not obtainable, therefore the value function is estimated for the strongest policy using reinforcement learning.

2 Results

In order to evaluate the results of the custom heuristic function the script `tournament.py` was run three times to control for random luck and the results were agregated.

2.1 Round 1

Table 1 shows the results of the first round of the tournament. AB Improved slightly outperformed the other functions, Center bias was the second while Edge bias and Aggressive attack tied for the third place.

Table 1: Round 1

Opponent	AB Improved	Center bias	Edge bias	Aggressive attack
Random	10 — 0	10 — 0	9 — 1	10 — 0
MM Open	7 — 3	6 — 4	6 — 4	6 — 4
MM Center	10 — 0	10 — 0	8 — 2	9 — 1
MM Improved	6 — 4	6 — 4	7 — 3	6 — 4
AB Open	6 — 4	4 — 6	4 — 6	5 — 5
AB Center	6 — 4	6 — 4	8 — 2	5 — 5
AB Improved	3 — 7	5 — 5	4 — 6	5 — 5
Win Rate	68.6%	67.1%	65.7%	65.7%

2.2 Round 2

Table 2 shows the results of the second round of the tournament. Edge bias which was the last in the previosu round clearly outperformed the competitors. Center bias was the second, Aggressive attack was the third and AB Improved, the winner of the previous round, was the last.

Table 2: Round 2

Opponent	AB Improved	Center bias	Edge bias	Aggressive attack
Random	10 — 0	10 — 0	9 — 1	9 — 1
MM Open	7 — 3	8 — 2	10 — 0	9 — 1
MM Center	10 — 0	10 — 0	9 — 1	9 — 1
MM Improved	7 — 3	7 — 3	8 — 2	5 — 5
AB Open	4 — 6	5 — 5	5 — 5	6 — 4
AB Center	5 — 5	5 — 5	5 — 5	7 — 3
AB Improved	4 — 6	4 — 6	6 — 4	3 — 7
Win Rate	67.1%	70.0%	74.3%	68.6%

2.3 Round 3

Table 3 shows the results of the third round of the tournament. Center bias tied Aggressive attack for the first place and AB Improved tied Edge bias for the third place.

Table 3: Round 3

Opponent	AB Improved	Center bias	Edge bias	Aggressive attack
Random	8 — 2	10 — 0	10 — 0	10 — 0
MM Open	5 — 5	8 — 2	8 — 2	8 — 2
MM Center	8 — 2	7 — 3	8 — 2	10 — 0
MM Improved	6 — 4	7 — 3	7 — 3	7 — 3
AB Open	6 — 4	6 — 4	3 — 7	3 — 7
AB Center	6 — 4	5 — 5	5 — 5	6 — 4
AB Improved	5 — 5	5 — 5	3 — 7	4 — 6
Win Rate	62.9%	68.6%	62.9%	68.6%

2.4 Aggregation of the three rounds

Table 4 shows the results of the aggregated results. Center bias performed the best, Edge bias tied Aggressive attack for the second place and the AB Improved was the last. Center bias strong performance in all rounds determined that it is the best heuristic function among of these analysed, nevertheless, the results were quite close and it is clear that many more rounds are needed to determine if it is truly better than the others with statistical significance.

Table 4: Total

Opponent	AB Improved	Center bias	Edge bias	Aggressive attack
Random	28 — 2	30 — 0	28 — 2	29 — 1
MM Open	19 — 11	22 — 8	24 — 6	23 — 7
MM Center	28 — 2	27 — 3	25 — 5	28 — 2
MM Improved	19 — 11	20 — 10	22 — 8	18 — 12
AB Open	16 — 14	15 — 15	12 — 18	14 — 16
AB Center	17 — 13	16 — 14	18 — 12	18 — 12
AB Improved	12 — 18	14 — 16	13 — 17	12 — 18
Win Rate	66.2%	68.6%	67.6%	67.6%

3   Conclusion and recommendations

All three of the introduced custom heuristics functions overall performed slightly better than the bechmark AB Improved function. Center bias heuristic function performed consistently and strongly over all of the three rounds taking the first or the second place and becomming the overall winner of the anaysed functions. Based on the facts that the Center bias heuristics function 1. outperformed AB Improved benchmark function, 2. outperformed the other bias heuristics functions and 3. performed consistently well in all round, showing little variation in results, it is the recommended custom heuristics function to use.