Saurabh Vyas
Udacity Artificial Intelligence Nanodegree
05-Feb-18

Evaluating Game Playing Heuristics

Project 2: Build a Game-Playing Agent

I evaluated three basic heuristics that can be used to score in Isolation game. All of these heuristics were used with alpha beta pruning.

1. AB_Custom (Own moves – Opp moves – distance from center) : This heuristic calculates the number of moves available to the player and then subtracts number of available moves to opponent and distance of the current move from center of the board. The basic idea is to maximize the number of moves available for the player and minimize the other two components. Distance from the center part tries to avoid the walls and opponent move part tries to make sure that opponent has less choices so that it eventually runs out of moves.
2. Difference of own moves and opponent moves (AB_Custom_2): This heuristic focuses on maximizing difference between own moves and opponent moves. This is actually a simpler version of the AB_Custom.
3. Weighted difference of own moves and opponent moves (AB_Custom_3): This option evaluates if applying a different multipliers on own moves and opponent moves and then subtracting them leads to a better function. For this heuristic, the different multiplier were tried such as:
    - OwnM – 1.5*OppM
    - OwnM – 0.5*OppM (found to be best performing in several trials)
    - OwnM – (OppM)**0.5

The following are the results of tournament.py. Since the results of the game could vary, I tried running tournament.py several times to ensure results are reliable.

```
                    **************************
                         Playing Matches
                    **************************

Match #   Opponent    AB_Improved   AB_Custom   AB_Custom_2  AB_Custom_3
                       Won | Lost   Won | Lost   Won | Lost   Won | Lost
   1       Random       9  |   1     10  |   0     10  |   0     9  |   1
   2       MM_Open      7  |   3      7  |   3      5  |   5     9  |   1
   3      MM_Center     9  |   1     10  |   0      8  |   2     9  |   1
   4     MM_Improved    8  |   2      9  |   1      8  |   2     5  |   5
   5       AB_Open      6  |   4      7  |   3      5  |   5     6  |   4
   6      AB_Center     6  |   4      5  |   5      4  |   6     6  |   4
   7     AB_Improved    5  |   5      6  |   4      5  |   5     6  |   4
--------------------------------------------------------------------------
         Win Rate:      71.4%        77.1%        64.3%        71.4%
```

```
                    **************************
                         Playing Matches
                    **************************

Match #   Opponent    AB_Improved   AB_Custom   AB_Custom_2  AB_Custom_3
                       Won | Lost   Won | Lost   Won | Lost   Won | Lost
   1       Random       9  |   1     10  |   0     10  |   0     10  |   0
   2       MM_Open      9  |   1      6  |   4      7  |   3     10  |   0
   3      MM_Center    10  |   0      9  |   1     10  |   0     10  |   0
   4     MM_Improved    7  |   3      9  |   1      8  |   2      8  |   2
   5       AB_Open      5  |   5      6  |   4      6  |   4      3  |   7
   6      AB_Center     5  |   5      7  |   3      6  |   4      6  |   4
   7     AB_Improved    4  |   6      6  |   4      7  |   3      4  |   6
--------------------------------------------------------------------------
         Win Rate:      70.0%        75.7%        77.1%        72.9%
```

```
                    **************************
                         Playing Matches
                    **************************

Match #   Opponent    AB_Improved   AB_Custom   AB_Custom_2  AB_Custom_3
                       Won | Lost   Won | Lost   Won | Lost   Won | Lost
   1       Random      10  |   0    10  |   0    9   |   1    10  |   0
   2       MM_Open      7  |   3     9  |   1    6   |   4    10  |   0
   3      MM_Center     7  |   3    10  |   0    9   |   1    10  |   0
   4     MM_Improved    6  |   4    10  |   0    7   |   3    9   |   1
   5       AB_Open      6  |   4     6  |   4    6   |   4    7   |   3
   6      AB_Center     5  |   5     5  |   5    6   |   4    4   |   6
   7     AB_Improved    5  |   5     3  |   7    5   |   5    4   |   6
-------------------------------------------------------------------------
         Win Rate:      65.7%        75.7%        68.6%        77.1%
```

	AB_Improved	*AB_Custom*	*AB_Custom_2*	*AB_Custom_3*
Round 1	71.4%	77.1%	64.3%	71.4%
Round 2	70.0%	75.7%	77.1%	72.9%
Round 3	65.7%	75.7%	68.6%	77.1%
Mean	**69.0%**	**76.2%**	**70.0%**	**73.8%**

Recommendation

The above results show that the third heuristic performs significantly better than the other two. This leads to the conclusion that an objective function on the below lines could be a very good choice:

F(x) = a*OwnMoves – b*OppMoves – c*(distance from center)

We can try to find the optimum values of ‘a’, ‘b’ and ‘c’ by simulation.

The reason why this function works well could be due to the following:

1. It tries to maximize the own moves
2. It also applies a penalty to those moves that give large number of options to the opponent and tries to avoid going towards walls or corners.
3. The penalties could be optimized further using simulations, which would lead to even better results.

Based on above it could be conclude that finding good heuristic function is a major part of any game strategy. A good heuristic is the one that is as close as possible to actual utility value.