Orlando Trejo November 27, 2017

AIND-Isolation: Heuristic Analysis

Three custom evaluation heuristic functions were developed and tested for the AIND-Isolation project. The purpose of evaluation function is to gauge the goodness of the children of a node. A successful evaluation function will frequently expand nodes that lead to the agent winning a game. In our isolation game, it was suggested in the lessons that an improved isolation function that took the difference between the number of moves remaining between the agent and opponent (own_moves - opp_moves) should capture the general trend of evaluating goodness of a node. This evaluation function is labeled **AB_Improved**. That is, expanding a node that results in a game board with more available moves for the agent than the opponent increases the chances of the agent wining. The purpose of this section to explore other evaluation functions and compare their performance.

The first custom evaluation function, **AB_Custom**, is a small modification of AB_Improved. In this function, the opp_moves are multiplied by a factor of 4 (own_moves - 4 * opp_moves), which increases the penalty the more moves an opponent has available. In other times, it is 4X worse for the opponent to have available moves. This function was shown to have an increased win rate average (Fig. 1) when compared to AB_Improved. Penalizing opponent moves by a factor of 4 is better than penalizing them by a factor of 2. Fig. 1 shows statistical behavior of running the evaluation functions multiple times (~10). AB_Custom is the preferred evaluation function at this point since it constantly results in a higher win rate as shown by its average wing rate of ~75%. Please see appendix for example tournament summary tables.

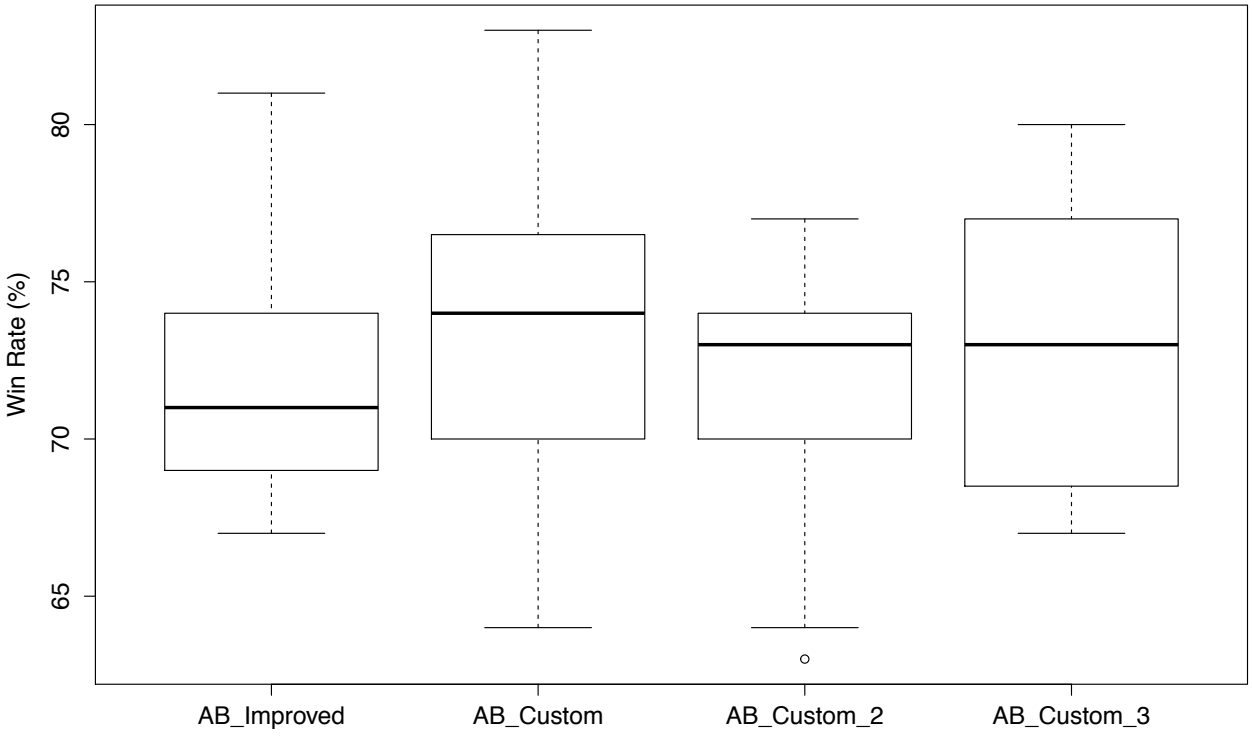

Figure 1. Box plot comparing different evaluation functions of AIND-Isolation project.

The second custom evaluation function, **AB_Custom_2**, takes the ratio between the difference of available moves and the total the moves available ((own_moves - opp_moves) / (own_moves + opp_moves)). This evaluation increases as the fraction of available moves for the agent increases. The evaluation decreases as the fraction of available moves for the opponent increases. As shown in Fig. 1, AB_Custom_2 has slightly higher average than AB_Improved but not markedly lower than AB_Custom. Like AB_Custom, it also has a lower minimum than AB_improved. This means that both custom algorithms may be susceptible to a similar type of board configurations that leads to incorrect expansion of nodes. **NOTE**: The ratio noted above will be referred to as **x/y** in the next paragraph.

In the final custom evaluation function, **AB_Custom_3**, the ratio x/y is inserted into a shifted sigmoid function (Fig. 2): 1 / (1 + exp(x/y - 2)). In this case, as the fraction of available increases for the agent the evaluation increases initially at a constant slope and once the fraction passes a threshold the evaluation will star to level off. Similarly, as the fraction of available decreases for the agent the evaluation decreases steadily until a threshold value. Afterwards, a decreasing fraction will result in a flatter evaluation curve. The reason for using a sigmoid is to capture in another way the probabilistic nature of winning as a function of available spaces. Since the ratio will most likely be lower than 1 in contested board configurations, then a shift in the sigmoid function towards more positive values showed an increase in the average win rate. This resulted in overall better performance than AB_Improved. However, it was still not possible to get a higher average win rate than AB_Improved.

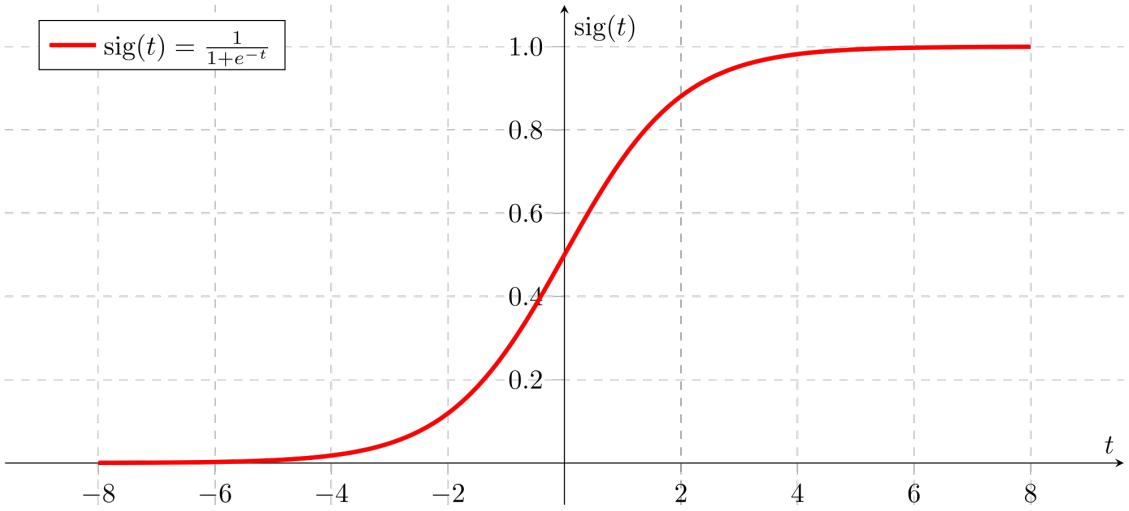

Figure 2. Plot of a sample sigmoid function.

Overall, all the three custom evaluation functions shows an average win rate higher than the AB_Improved. AB_Custom is chosen as the final evaluation function since: 1) it has a markedly higher average win rate than the rest; 2) it is an easy function to compute so it does not accrue a high computational cost or memory allocation; and 3) it is easy to implement and understand intuitively. The appendix below contains summary table examples. However, more improvements may be done so it can at least have a higher lower win rate value. It may be

possible to continue mix and matching these evaluations functions to come with a better performance. For instance, the factor of 4 can be incorporated into the x/y ratio. Furthermore, my current understanding of the game and underlying algorithm prevents me from probing deeper into the nature of the evaluation function. I look forward to gaining more understanding in this nanodegree to be able to answer such evaluation function design questions more strategically.

Appendix

Table 1. Summary of results comparing performance of above evaluation functions.

Match #	Opponent	AB_Improved		AB_Custom		AB_Custom_2		AB_Custom_3	
		Won	Lost	Won	Lost	Won	Lost	Won	Lost
1	Random	8	2	8	2	9	1	9	1
2	MM_Open	8	2	6	4	9	1	9	1
3	MM_Center	9	1	10	0	8	2	9	1
4	MM_Improved	7	3	8	2	8	2	8	2
5	AB_Open	4	6	3	7	5	5	7	3
6	AB_Center	9	1	7	3	9	1	9	1
7	AB_Improved	4	6	6	4	5	5	5	5
	Win Rate:	70.0%		68.6%		75.7%		80.0%	

Table 2. Summary of results comparing performance of above evaluation functions.

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	10	0	10	0
2	MM_Open	8	2	5	5	8	2	8	2
3	MM_Center	9	1	10	0	10	0	10	0
4	MM_Improved	8	2	6	4	5	5	9	1
5	AB_Open	6	4	5	5	4	6	4	6
6	AB_Center	7	3	5	5	7	3	5	5
7	AB_Improved	4	6	4	6	5	5	5	5
	Win Rate:	74.3%		64.3%		70.0%		72.9%	

Table 3. Summary of results comparing performance of above evaluation functions.

Match #	Opponent	AB_Improved		AB_Custom		AB_Custom_2		AB_Custom_3	
		Won	Lost	Won	Lost	Won	Lost	Won	Lost
1	Random	9	1	9	1	10	0	8	2
2	MM_Open	7	3	9	1	7	3	8	2
3	MM_Center	9	1	10	0	10	0	6	4
4	MM_Improved	6	4	8	2	7	3	6	4
5	AB_Open	5	5	7	3	6	4	7	3
6	AB_Center	7	3	5	5	4	6	7	3
7	AB_Improved	5	5	6	4	6	4	5	5
	Win Rate:	68.6%		77.1%		71.4%		67.1%	

Table 4. Summary of results comparing performance of above evaluation functions.

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	8	2	9	1
2	MM_Open	8	2	8	2	8	2	7	3
3	MM_Center	9	1	9	1	10	0	9	1
4	MM_Improved	8	2	6	4	6	4	9	1
5	AB_Open	7	3	5	5	7	3	4	6
6	AB_Center	5	5	7	3	6	4	8	2
7	AB_Improved	8	2	6	4	6	4	8	2
	Win Rate:	78.6%		72.9%		72.9%		77.1%	

Table 5. Summary of results comparing performance of above evaluation functions.

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	10	0	10	0
2	MM_Open	8	2	7	3	8	2	7	3
3	MM_Center	8	2	9	1	9	1	10	0
4	MM_Improved	7	3	7	3	8	2	7	3
5	AB_Open	4	6	7	3	3	7	7	3
6	AB_Center	7	3	6	4	6	4	6	4
7	AB_Improved	7	3	6	4	5	5	7	3
	Win Rate:	72.9%		74.3%		70.0%		77.1%	

Table 6. Summary of results comparing performance of above evaluation functions.

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	9	1
2	MM_Open	8	2	8	2	10	0	5	5
3	MM_Center	10	0	9	1	7	3	10	0
4	MM_Improved	7	3	7	3	8	2	8	2
5	AB_Open	4	6	5	5	4	6	6	4
6	AB_Center	3	7	7	3	4	6	5	5
7	AB_Improved	5	5	7	3	3	7	6	4
	Win Rate:	67.1%		75.7%		64.3%		70.0%	