Micah Brighton
James Bury
Spencer Siegel

Assignment 2

Model-Based

1. Changing the exploration probability impacted the variance of the outcomes quite a bit. It seemed that there were a wider range of recommended policies when using a small exploration probability. In this scenario, in a given state, certain aims may not be attempted (or are hardly attempted) so it is hard to get a representative utility for that state. The state/action/state probabilities are likely to be more inaccurate. Additionally, as noted above, the variance in the utilities for each state is much greater with lower exploration probabilities. Lastly, using a lower exploration probability typically results in faster convergence. With less exploring, the model takes advantage of what it knows to be effective and does not spend time exploring states that it deems ineffective. This results in faster convergence

2. For model-based, we passed an argument into the function called “threshold.” We defined “threshold” as the convergence threshold for utility. If any state's utility changes by more than the threshold value then the model keeps learning. We passed a second value as an argument called “iterations,” which checks how many iterations in a row the utility has changed by less than threshold. Once there are 3 iterations, we decided to stop learning.

3. Changing the discount value impacted the range of outcomes for model-based active RL. A low discount value signifies that the utility of each state is mainly comprised of the reward at that state. Each state has a reward of 1 (besides “In”). In this scenario, a low discount value causes the utilities to not be representative of how close the ball is to reaching the final state (“In). Therefore, the learning and policy recommended in this scenario are flawed. This leads to a wide range of recommended policies, many of which would not make sense. Using a higher discount value leads to a more accurate utility for each state. This is because it puts a higher weight on the component that represents how close the ball is to the “In” state. Therefore, a higher discount value has a narrower set of optimal policies and leads to more logical utilities for each state.

4. Changing epsilon had a major impact on the outcome of model-based active RL. Using a high value for epsilon results in much quicker convergence. Therefore, the probabilities have not converged to the true probabilities. This impacts the results because the utilities and recommended policy will be inaccurate and misinformed. A larger value of epsilon also results in a higher variance in outcomes (both in probabilities and policies). This is because the model converges so quickly with little information. So there is a wide range of outcomes that it will see as optimal. When using smaller values of epsilon, the

outcomes are much narrower. The model takes longer to converge, but is more accurate in its probabilities and utilities. Therefore, it recommends a more reliable policy (although it takes longer to get there).

Model Free:

1. Changing the initial value for exploration resulted in differing amounts of variance in the utilities for each state/action pair when the program was run multiple times. With a starting exploration value of 0.25, the largest utility (most shots required to get to state “In” on average for a state/action pair) ranged from 5.5 to 8 when the program was run four times. For an initial exploration value of 0.5, the largest utility ranged from 5 to 7. For 0.75, the range was between 3.8 and 5, and for a value of 1.0 the largest utility ranged from 4 to 5. This shows that as the initial exploration value is increased from 0 to 1, the resulting state/action pair utilities become more consistent when the model is run multiple times.

2. For model free, we decided to stop learning only once all three of the following occur: no new states were discovered in the last run, the sum of all the changes in state/action utilities was less than our threshold of 0.001, and there were no remaining undiscovered state/action pairs.

3. N/A

4. Changing the value of epsilon resulted in higher variance and quicker convergence in our model free algorithm. When running the program eight times with each epsilon in {0.001, 0.01, 0.1, 0.5}, we came up with these values for average variance in state/action utility results:

   {0.001: 0.33289395411130673,
   0.01: 1.9041888219307572,
   0.1: 1.3886248886248886,
   0.5: 14.833333333333332}

   The difference between the highest average variance with epsilon 0.5 and the lowest average variance with epsilon of 0.001 is 14.833 - 0.332 = 14.501. Additionally, holding epsilon at 0.001 took an average of 103 iterations, while using an epsilon of 0.5 only took on average 15.75 iterations to converge. This shows that as we increased epsilon from 0 to 1, it would converge more quickly and produce higher variations of utilities for each state/action pair between runs.