Ji Hoon Kim September 28, 2014

CS440 MP1 Part 2: Experimentation

1) Problem 1: World Without Thief

   a) After simulating the Q-Learning Agent with the provided settings for the problem, it seemed like the agent was unable to properly navigate the environment. It received an initial reward of -8.5 for the first episode, but the following rewards all remained at 0.

   b) After setting the ε = 0.1, it seemed as if there was an improvement in what the agent was capable of. The episodes running the simulator generated many rewards in excess of 150 points and many others that were slightly below, but its total rewards were high in the positive.

   c) After setting the ε to various, larger values, the total rewards seemed to decrease quite a bit. Increasing ε to be 0.3 kept the rewards in the positive for the most part but once I reached ε = 0.7, it seemed to be solidly into the negative totals for the rewards. Increasing to ε = 0.9 did not seem to show a significant difference from when ε = 0.7.

2) Problem 2: World With Thief

   a) The performance of the agent is not great. The total rewards generated from the simulation are in the negatives, somewhere between -50.0 through -100.0. To validate the performance of the agent, I ran it through the Policy Simulator, but it did not perform very well as the agent seemed to get stuck against the north edge of the wall, unable to circumvent the thief.

   b) The performance seemed to slip even more still. Even though the agent knew of the Thief, it returned many total rewards that were less than -250.0. The randomness caused by the epsilon may have been too much as it seems to have made the agent randomly go straight into the Thief leading to the lower reward totals. Ultimately, the values did not seem to converge, but rather fluctuate pretty sporadically for the 1000 episodes.

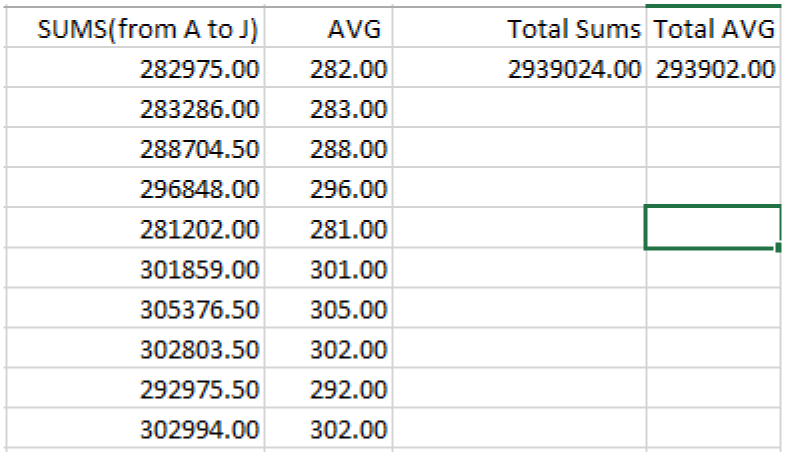
   c) After many test runs, it looks like the best learning rate = 0.45 and ε = 0.001. How I came up with these values were as follows:

      i) First I decided that the best learning rate and ε would provide me the highest total rewards per episode.

      ii) Secondly I started by randomly plugging in values just to determine a general range for each values. This showed me that the learning rates closer to 0.50 and ε = 0.001 provided the highest total rewards.

      iii) From there, I started to take the values and the total sums (sum of total rewards from all episodes) and averages (total rewards from all episodes/1000) over the span of 10 simulations with learning rates of +- 0.1 at intervals of 0.01.

      iv) After taking all the numbers and determining the best 5, I took those values and totaled the rewards and took the average of the simulations and decided that the provided values for the learning rate performed the best.



SUMS(from A to J)	AVG	Total Sums	Total AVG
282975.00	282.00	2939024.00	293902.00
283286.00	283.00		
288704.50	288.00		
296848.00	296.00		
281202.00	281.00		
301859.00	301.00		
305376.50	305.00		
302803.50	302.00		
292975.50	292.00		
302994.00	302.00		

Figure 1 Showing results for learning rate = 0.45 and ε = 0.001.

3) <u>Problem 3: Plotting Results from World With Thief</u>

a) When first plotting the reward achieved, the progression of the agent was almost converging but a good amount of fluctuation was still visible. The remaining 9 attempts also showed varying degrees of fluctuations in total rewards even if the values it fluctuated between were increasing.

The most interesting thing happened when the 10 simulations were averaged out and results plotted on the chart. The convergence of the performance is easily noticeable as the level of fluctuations is greatly decreased. This seems to imply that over time, the performance will properly converge at an optimal solution.

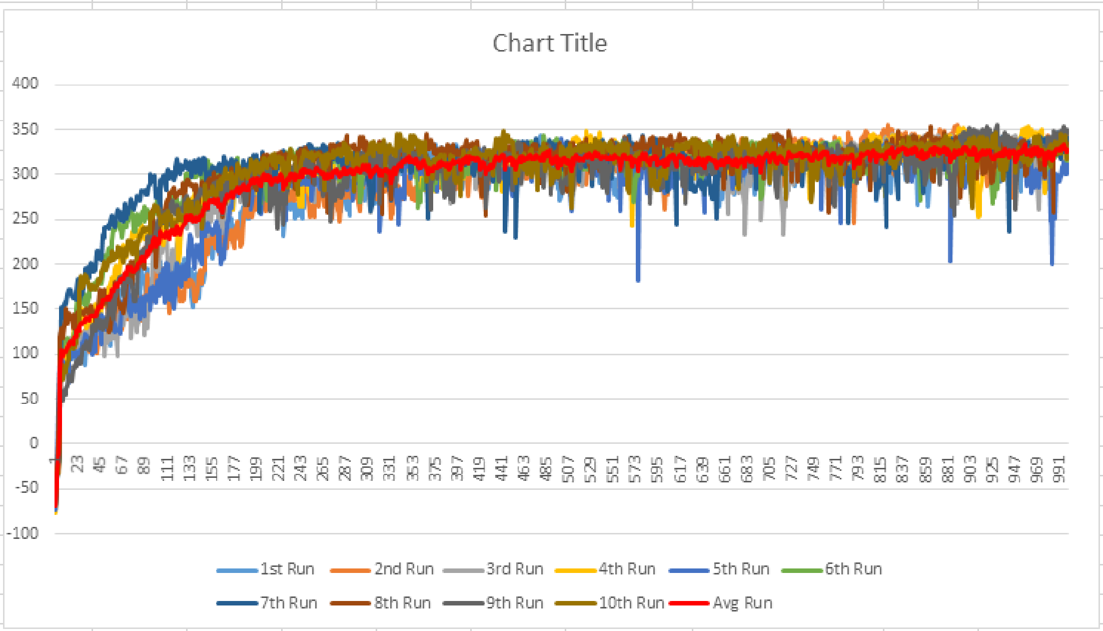


Figure 2 Shows the chart of 10 simulations and its average plotted in a chart. Notice the red line representing the average.