CS 440

MP 1 Part 2
nabachm2 (Nicholas Bachmann)

1: a: For $\epsilon = 0.0$ we see that the agent doesn't make any progress towards the goal. Instead it just travels north and repeatedly runs into the wall. This is most likely because any slipping that occurs in a simulation results in a negative reward, so the robot never explores that state again, even if that state is necessary to reach the goal.

b: For $\epsilon = 0.1$ we see that the agent usually preforms optimally. When traveling its avoids the most slippery spots and only passes through the slippery areas the least amount of times possible when delivering the package (3 times). This is because the agent "learns" that the slippery spots result in a bad rewards, however, it still has enough randomness to find the goal, and receive a large rewards. Occasionally, every couple of simulations, the bot will make no progress towards the goal, and gets stuck. I believe this is due to the randomness of the bot, and the world requires a large amount of things to go right to reach a positive reward.

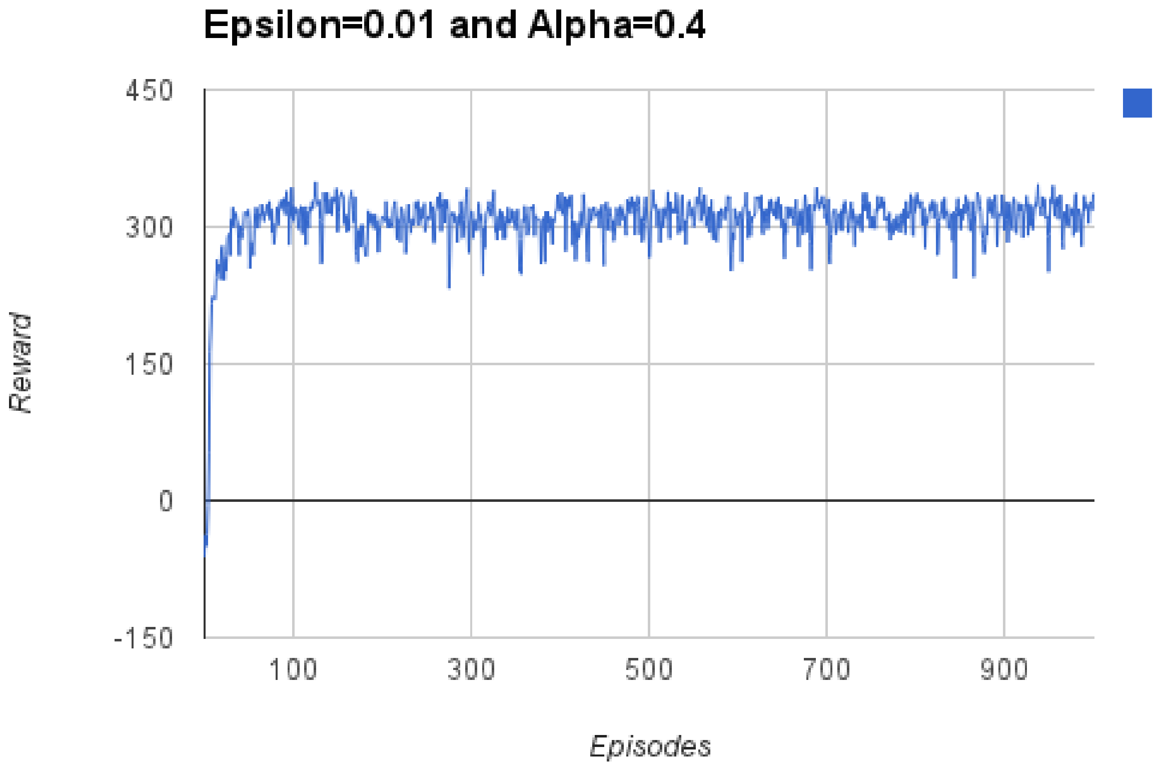
c: For $\epsilon = 0.6$ we see that the agent almost never reaches the goal state, and when it does it is usually not optimal. This is because the agents stops learning, and more often then not, relies on randomness to determine its movement. Given the amount of things that need to go right for the agent to achieve the goal state (i.e. hit 2 specific spots in a grid of 25 states, there is a very small chance the agent will get a positive reward.

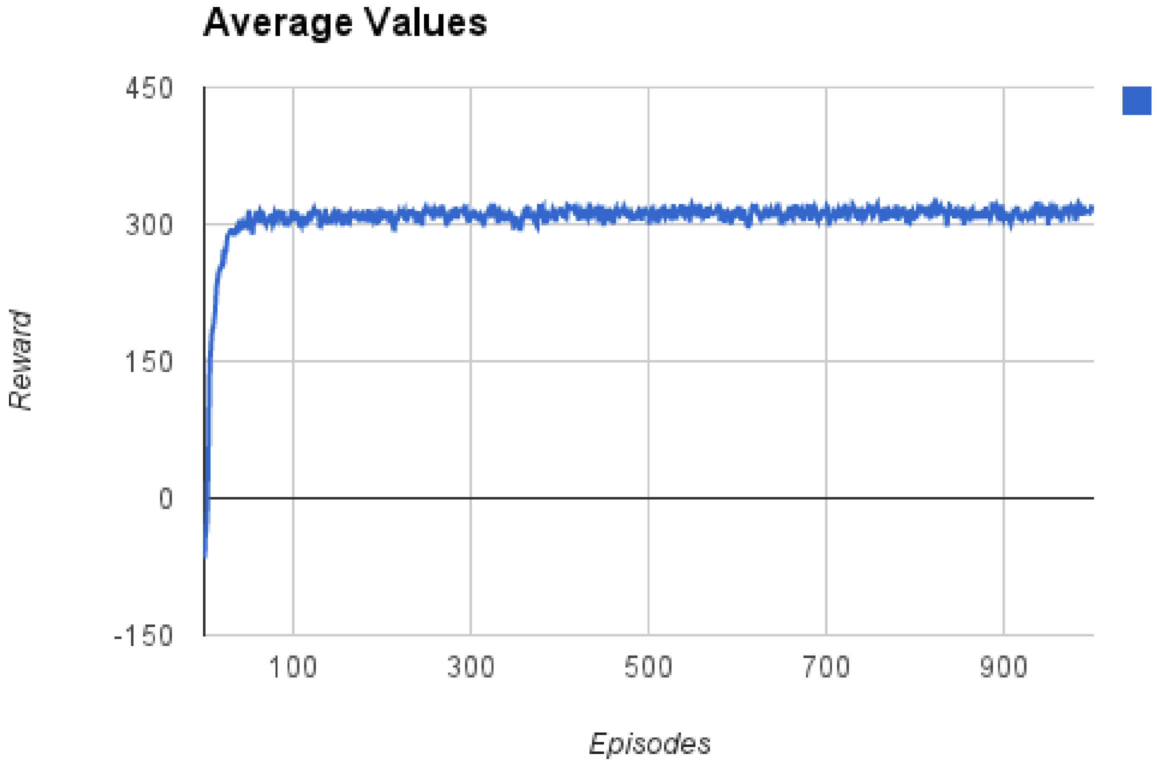
2: a: For $\epsilon = 0.05$ and KnowsThief set to no, we see that the agent doesn't make any progress towards the goal. Instead it just travels north and repeatedly runs into the wall. This is most likely because in every scenario the thief steals the agent's package, since the agent and thief are in a very confined space and their chance of meeting is very high. Since the agent doesn't know where the thief is, it has no way to learn how to avoid it. This results in a negative rewards for any movement towards the right side of the board where the thief is.

b: For $\epsilon = 0.05$ and KnowsThief set to yes, we see that the agent agent usually preforms optimally. It dodges the thief and avoid the slippery areas. This is because the agent can correctly learn to avoid the thief (since it knows where it is).

c: I determined the best range for values to be $\epsilon \approx 0.01$ and learning rate $\approx .3$ The general trend was that as ϵ was decreased the max reward increased, until it peaked around 0.01. With lower values of epsilon it would take the agent much longer to converge (almost 150 episodes when ϵ is 0.001). Learning rate seemed a little harder to pinpoint, since it was often depended on the value for ϵ, but generally speaking, as learning rate was decreased (from the original value of .9), the total reward increased. Reward would increase until learning rate reached .4, where it plateaued until it became less than 0.1, in which case the reward began decreasing.

3: Plotting the data we see that our agent converges to hits maximum reward very quickly (within the first 20 episodes). Further, his reward stays at about the same level, with small fluctuations between episodes. In the average we see this convergence more sharply, and the noise seen in the single reading is canceled out such that there is very little variation between episodes. Here are the plots


Epsilon=0.01 and Alpha=0.4
450
300
150
0
-150
Reward
100
300
500
700
900
Episodes


Average Values
450
300
150
0
-150
Reward
100
300
500
700
900
Episodes