Neelan Coleman's
CS 440 / ECE 448: Introduction to AI
MP 1 (part 2)

Due Date: 12 am Monday, September 28, 2014

For this experiment I used the QLearningAgent.java provided by the Ta's for simplicity.

Assignments must be submitted via Compass 2g, in PDF. You are strongly en couraged to edit your answers in LaTeX. Do not submit handwritten solutions.

Problem 1. In WorldWithoutThief

(a) With the parameters set for learningRate = 0.1, epsilon = 0.0, and discountFactor = 0.9. I ran the simulation for 1,000 episodes having 10,000 step each. I noticed that the agent never acquired any positive rewards. Actually all rewards were zero. In the simulation over the resulted policy, the agent seems to pick a direction, "up" for example, and continue to do that action over and over again. I believe this is due to the lack of randomness in the configuration. The agent is not exploring the world enough. dded to the queue in alphabetical order whenever the order is

(b) after setting epsilon to 0.1, I noticed that in each episode the agent acquired positive rewards. This is because the agent began to explore the world more often, which led to learning. The resulted policy still seemed to have the agent alternate between several states indefinitely. This is likely due to not having enough steps in the simulation process.

(c) With a larger epsilon value (0.5) I noticed that the agent acquired very negative rewards. I believe this is due to too much exploration. The agent basically doesn't learn from his mistakes. He just explores all options and since the world is fairly negative, he acquires mostly negative rewards. In the policy simulator, he actually delivered one package before getting stuck in an infinite loop.

Problem 2. In WorldWithThief

(a) After setting epsilon to 0.05. I noticed that the rewards after each episode seemed to get better over time. They started negative and ended greatly positive. After about 30 episodes the rewards started to reach the hundreds. Also the consistency of consecutive high rewards increased as the number of episodes increased. The increase in reward is due to a balance of exploration and greediness. The greedy algorithm helps us improve as we explore the world more. The agent avoids negative rewards and goes for the most positive ones. The small epsilon factor prevents us from being "too greedy" and forces us to eventually explore the world. I believe this is why we see the gradual increase of rewards. I should also note that the negative start might be due to not knowing where the thief is in the world. Though the agent kind of learns implicitly to avoid the thief over time.

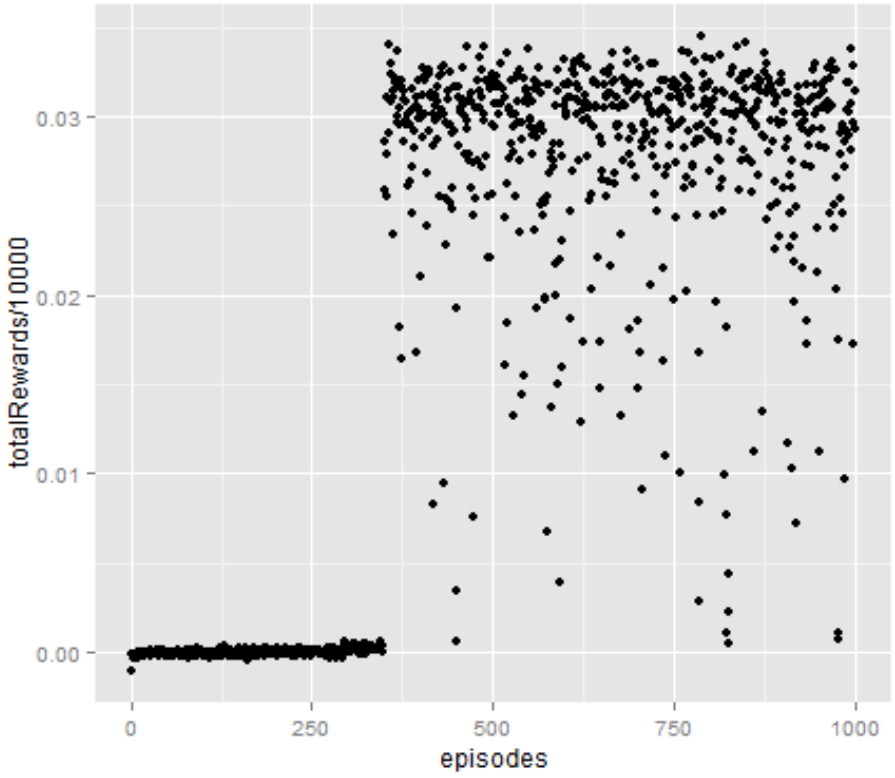
(b) after configuring the simulation to allow for the agent to be aware of the thief's position. I noticed that it took longer for the rewards to become positive. It took about 75 episodes as compared to the 30 episodes in part a. I think this is because we simply have many more states now that the thief's position is known. So it would make sense for the agent to take longer to explore the world and converge toward an optimal policy.

(c) To investigate the best epsilon and learning rate. I first fixed the epsilon value at 0.05. I then chose different values for the learning rate. It seemed like anything lower than 0.005 for the learning rate did not actually converge to an optimal policy. The rewards remained negative or close to zero for all episodes. I noticed that at 0.05. I observed the most positive rewards and the rewards jumped to "very positive" values early on. It tested values around 0.05 but they didn't seem as good. I also tested values much greater than 0.05, like 0.8, 0.1. They seemed to do well around 0.1 but the rewards were not as positive as with a learning rate of 0.05. At high learning rates like 0.8 the rewards were pretty negative. //After I found that 0.05 was a pretty good candidate for the learning rate. I fixed the learning rate as 0.05 and tinkered with epsilon. I found similar results for epsilon. Values less than 0.001 seemed consistently negative. And values around 0.01 seemed to get the best rewards. So my final choice for the parameters are: learningRate=0.05 and epsilon=0.01.

Problem 3. Using a learning rate of 0.05 and an epsilon of 0.01 steps 10,00 and 1,000 episodes.

(a) I observed that there was a pretty sharp jump to very positive

Figure 1: Grid World



rewards around 250 to the 300th episode.

(b) Surprisingly, the average of ten simulations makes for a more smooth curve. Should this be a surprise though... hmm. Not sure of what specific, underlying mathematics would guarantee a smooth curve in this case, but it seems like the average of the Expected Discounted Rewards tend to smoothly increase with increasing episodes. That's a property that we'd hope for in learning. It seems intuitive, but I'm not sure if it's neccessarily guaranteed with the qlearning algorithm.

Figure 2: Grid World