Eric Lee
eclee2
CS440
Fall 2014
09/29/2014

MP 1, Part 2: Experimentation

1.) In WorldWithoutThief

a.) The robot gets stuck at 0.0 reward and will never progress past that. This is because when the robot traverses the map, he will never choose a lesser utility square. This means that the robot will never try the negative reward spaces (slippery spaces), because the 0.0 reward squares are the better choice and there is no exploration built in. Since all of the positive rewards are on the other side of the slippery spaces and require the robot to cross, the robot will never learn about the reward on the other side and never achieve a reward greater than 0.0.

b.) Since the robot now has exploration as part of its learning schema, the robot can now progress to the other side of the board. However, since the exploration chance is so low (.1) the robot stays with negative reward for many runs until it finally discovers the high reward squares by chance. Once this happens, the reward accumulated instantly increases to a much larger amount (100+) and almost every subsequent episode now attains a high reward. There is also a chance that the robot attains a lower reward than expected because the small exploration factor leads them to slippery spaces.

c.) The robot now only chooses the highest utility space 50% of the time. Because of this, it explores the board faster, however the robot has very sporadic movement. Since it will only choose the logical space half the time, the robot makes very random choices, which often end up being very bad choices that lead to negative reward. This leads the robot to run into very negative rewards on each episode. The robot is learning a good utility schema for the spaces, but it doesn't utilize it enough to attain good results.

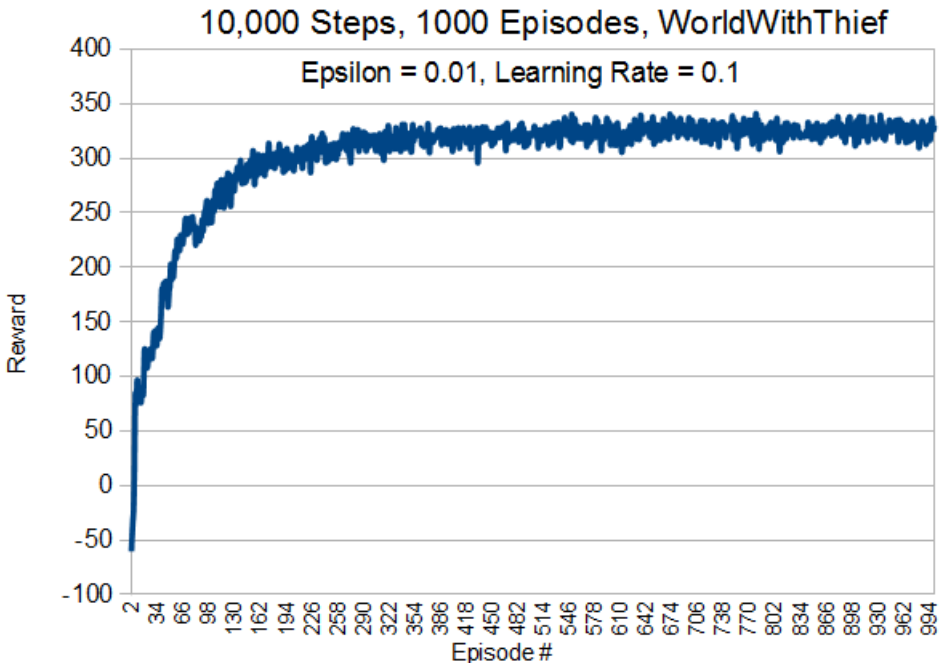
2.) In WorldWithThief

a.) This reward outcomes are very poor. This happens since the robot does not even know about the real reward of the thief space (it doesn't even know it exists) and it can not accurately predict utility of spaces to create an optimal policy. The robot will land on the thief space and incur negative reward, but will never learn from it since it can not detect it. The robot hovers around a -10 reward outcome.

b.) Now that the robot knows that the thief is there, it can avoid that space when calculating best action. It can also take the thief space's high negative reward into account when calculating utilities. This allows the robot to avoid the thief almost all of the time, and avoid negative reward. Because of this, the robot finds an optimal policy relatively quickly and achieves a high reward after about 10-15 episodes.

c.) The best learning rate happens with a low epsilon value around 0.01 and a low learning rate of around 0.1. By fluctuating the learning rate, I could quickly see that a learning rate too high would exaggerate the variation in reward between episodes, and limit the robots ability to converge to a stable policy. By choosing 0.1, the robot still has time to learn the best utility values of the spaces because of the high amount of steps and episodes, and can also develop a stable utility value that doesn't fluctuate. Then by altering the epsilon value, it became clear that the high step number and episode number made a low epsilon value advantageous. The low epsilon value limits the robots exploration rate, but since there are so many steps and episodes, the robot will still explore the board to learn an optimal policy in time. Then the advantage of a low epsilon comes once the robot has found an optimal policy. The robot will make a smart choice a large majority of the time, limiting the amount of negative rewards incurred by chance. These settings achieve a stable reward around 340.0 by the 500th episode.

3.)



This is the plot of the discounted reward from every episode from 1 to 1000. You can see that the reward rapidly increases, but then reaches a limit at around 340.0, approaching around the 200th episode and leveling off around the 350th episode.