Jui-Ting Hsu

CS 440 MP1 Part 2

Sept 28, 2014

1. In **WorldWithoutThief**
2. I observed that the reward is 0.0 for all the episodes. In the PolicySimulator, the robot gets stuck. Since epsilon is set to be 0, we are following a pure-greedy policy. All actions have other than crossing the slippery floor would have a utility of 0. Crossing the slippery floor would have a negative utility, so the robot never crosses the slippery floors, thus never delivering the packages.
3. There are some negative rewards in the first few episodes. After that, even though the rewards sometimes fall or sometimes rise, it’s always a positive reward for each episode. In the PolicySimulator, the robot was able to successfully deliver the packages with the misfortune of slipping on the slippery floors from time to time. The robot was able to function correctly because it is now epsilon-greedy instead of pure greedy. That is, for each action, there is a possibility of exploring the unknown for a potentially greater reward instead of just following the highest immediate utility. This would be effective because the reward in this scenario is not gained immediately but rather gained through a series of actions.
4. I tried epsilon = 0.5 and observed that more episodes had a negative reward. In the PolicySimulator, the robot became less efficient. Over time, it still earn rewards, but in a significantly slower pace than that in 1b. This is because the need of random exploration decreases over time, since we’ve collected more and more information about the world. Therefore, the sometimes-random action became less and less worthy, since the policy became better and better.
5. In **WorldWithThief**
6. The rewards for all the episodes are negative. In the policy simulator, the robot never leaves the company. Since it is not aware of thief position and is trying to avoid the slippery floor, there is only one way to cross the column. It is very possible to encounter the thief while delivering the packages because the robot is not aware of the position of the thief. So overall, the reward for delivering is negative, because the thief keeps on stealing the packages from the robot. Therefore, the agent then finds it better to do nothing and get 0 rewards instead of trying to deliver the packages and receive negative rewards.
7. The first few episodes might have negative rewards, but no negative rewards were observed after that. The robot was always able to deliver the packages in the PolicySimulator. The robot was able to avoid the slippery floor and wait for the thief to get out of its way when it’s carrying package(s). After delivering both packages, the robot doesn’t care if it crosses slippery floors. The robot was able to wait for the thief to get out of its way because it is aware of the position of the thief.

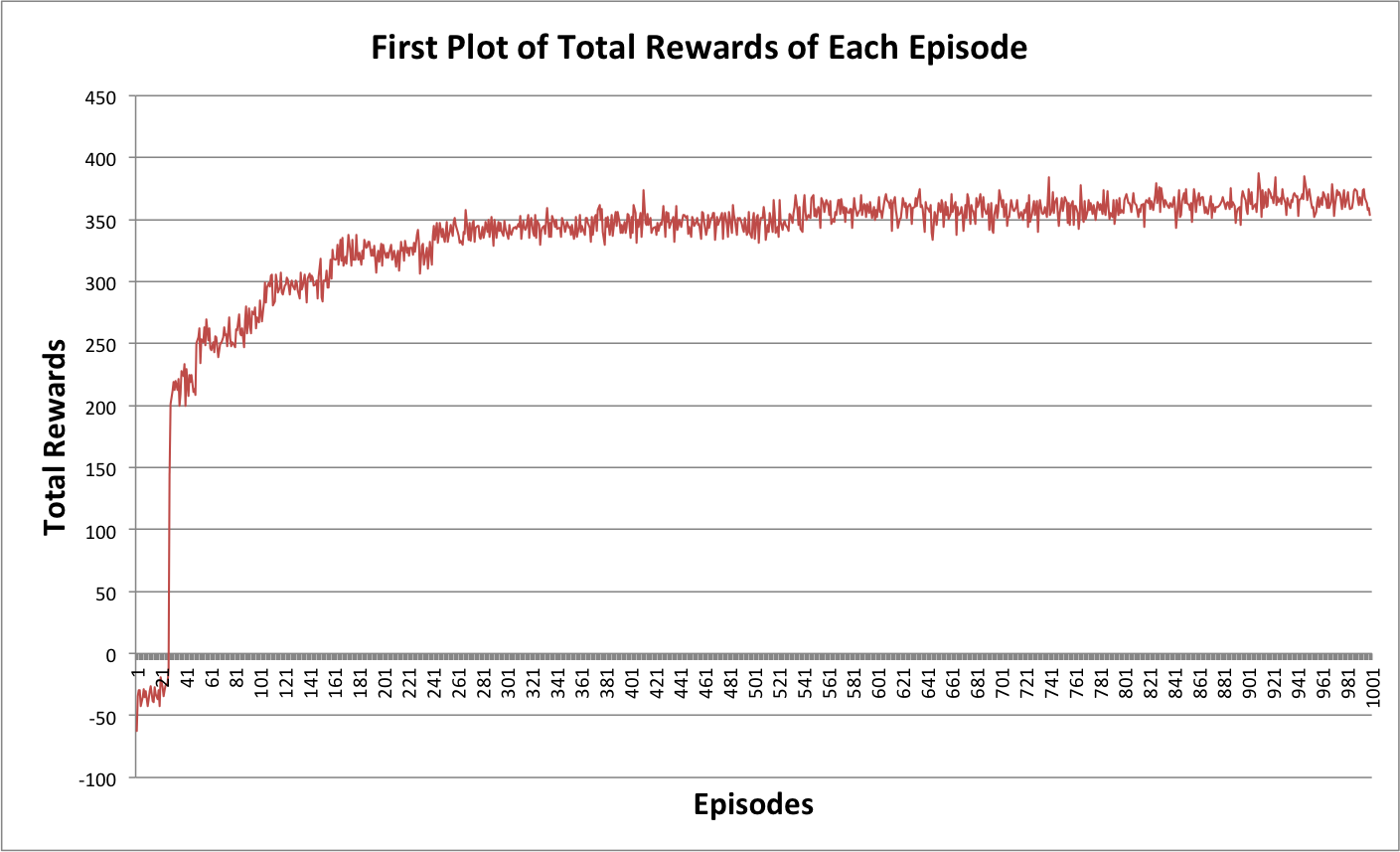
|  |  |
| --- | --- |
| epsilon | Average rewards over 1000 episodes  (with learning rate = 0.1) |
| 0.05 | 272.276 |
| 0.025 | 323.4275 |
| 0.0125 | 340.162 |
| 0.00625 | 330.742 |
| 0.003125 | 308.11 |

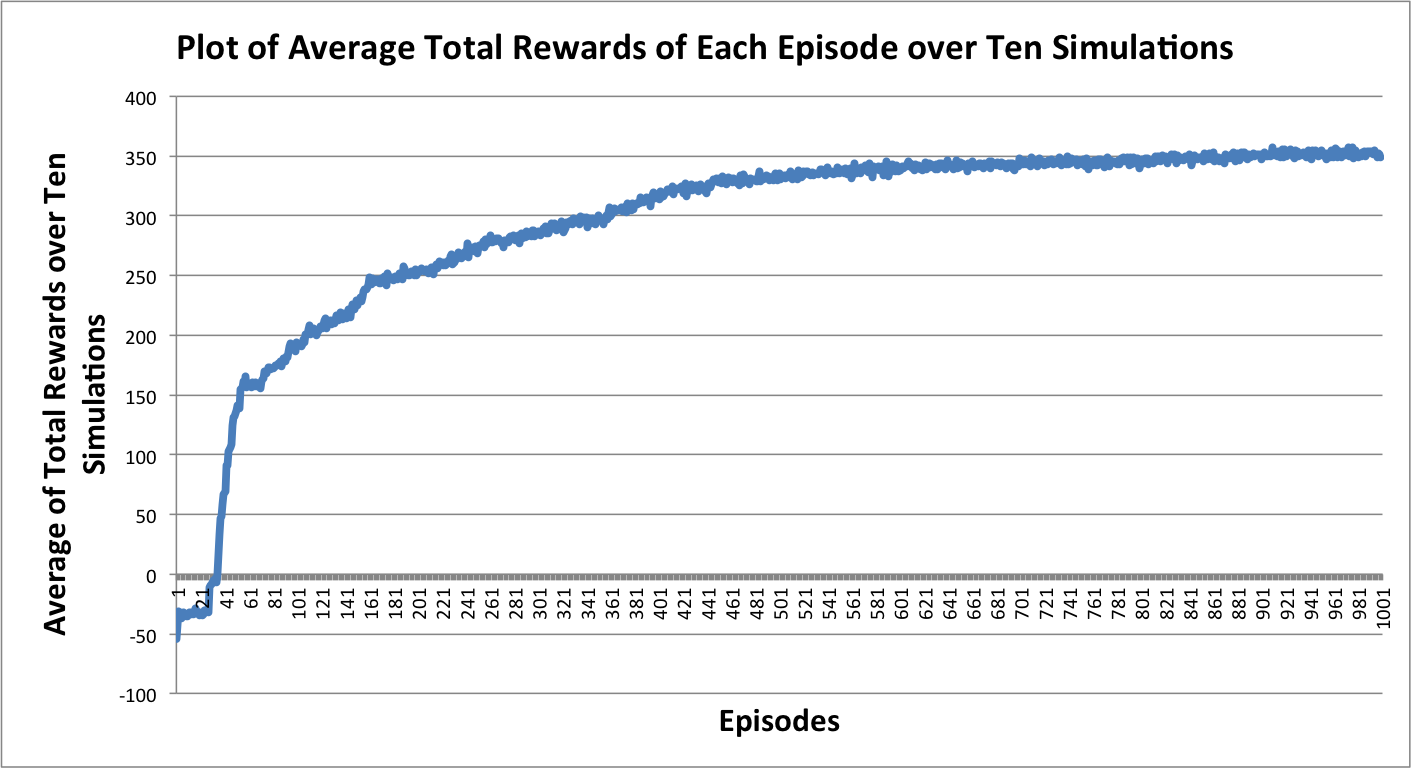
The best epsilon is about 0.0125.

|  |  |
| --- | --- |
| Learning rate | Average rewards over 1000 episodes  (with epsilon = 0.05) |
| 0.1 | 266.671 |
| 0.9 | 180.4105 |
| 0.5 | 233.249 |
| 0.05 | 276.3135 |
| 0.025 | 273.127 |

The best epsilon is about 0.05.

As a conclusion, I think that smaller epsilon and learning rate lead to a better agent. However, the effect of decreasing epsilon and learning rate decreases over time. Epsilon and learning rate should be small but not too small.

3. 



The plot of the average total rewards over ten simulations is similar but different to that of a single simulation. The difference is a result of the randomness of epsilon, the thief, and the boundary condition when searching for a best-utilized action. The similarity is because the agent is really learning and improving significantly for a period of time in the beginning by exploring the world. The result would be even better if the epsilon decays over steps since the need for random-epsilon move to explore the world decreases over time.