Karthik Gudipati

MP1 Part 2: Experimentation

---

1) In *WorldWithoutThief*

a) When I set the discount factor to 0.9, the rate to 0.1, and the epsilon to 0.0, I observed that for a 1000 episodes and 10000 steps, the majority of the rewards were 0.0. In fact, only the first episode had a rewards of -19.0 while the other 999 episodes had a reward of 0.0. Because our epsilon is 0.0, we will never allow for a random integer to be chosen for our actions. Thus, this leads to no exploration on part of the agent.

b) When I set the discount factor to 0.9, the rate to 0.1, and the epsilon to 0.1, I observed that for a 1000 episodes and 10000 steps, the rewards shot up incredibly. Now I observe rewards ranging from negative numbers to triple digits as far as 250. Because we have a small epsilon, of 0.1, our randomness is also not that large. However, because of this we receive a high performing agent. If we set our epsilon too large, the agent would chose very random action which lead to a poor performance. I notice that the agent did not start to get better, over negative numbered rewards, until around episode 22. From there on though the rewards kept increasing and eventually, towards episode 1000, the rewards were in the triple digits at a higher rate than before.

c) With a higher epsilon, around 0.5, the agent was very random in its decision of actions. Because of that its performance was very low and didn't even hit a triple digit reward. Even at an episode of 1000, the reward was -29.0 meaning that it was difficult for the agent to learn with the randomness in its decisions.

---

2) In *WorldWithThief*

a) When setting the epsilon to 0.05 and that the agent does NOT know where the thief is at, the performance is very low. This is mainly due to the fact that the agent doesn’t know the location of the thief. Although the agents decisions in actions were very random, with a low epsilon, the fact that the agent did not know where the thief was led to the agent running into it and resulting in constant negative rewards. The agent needs a better understanding of its world in order to perform better, which caused the agent in the world without thief to perform better with a low epsilon.

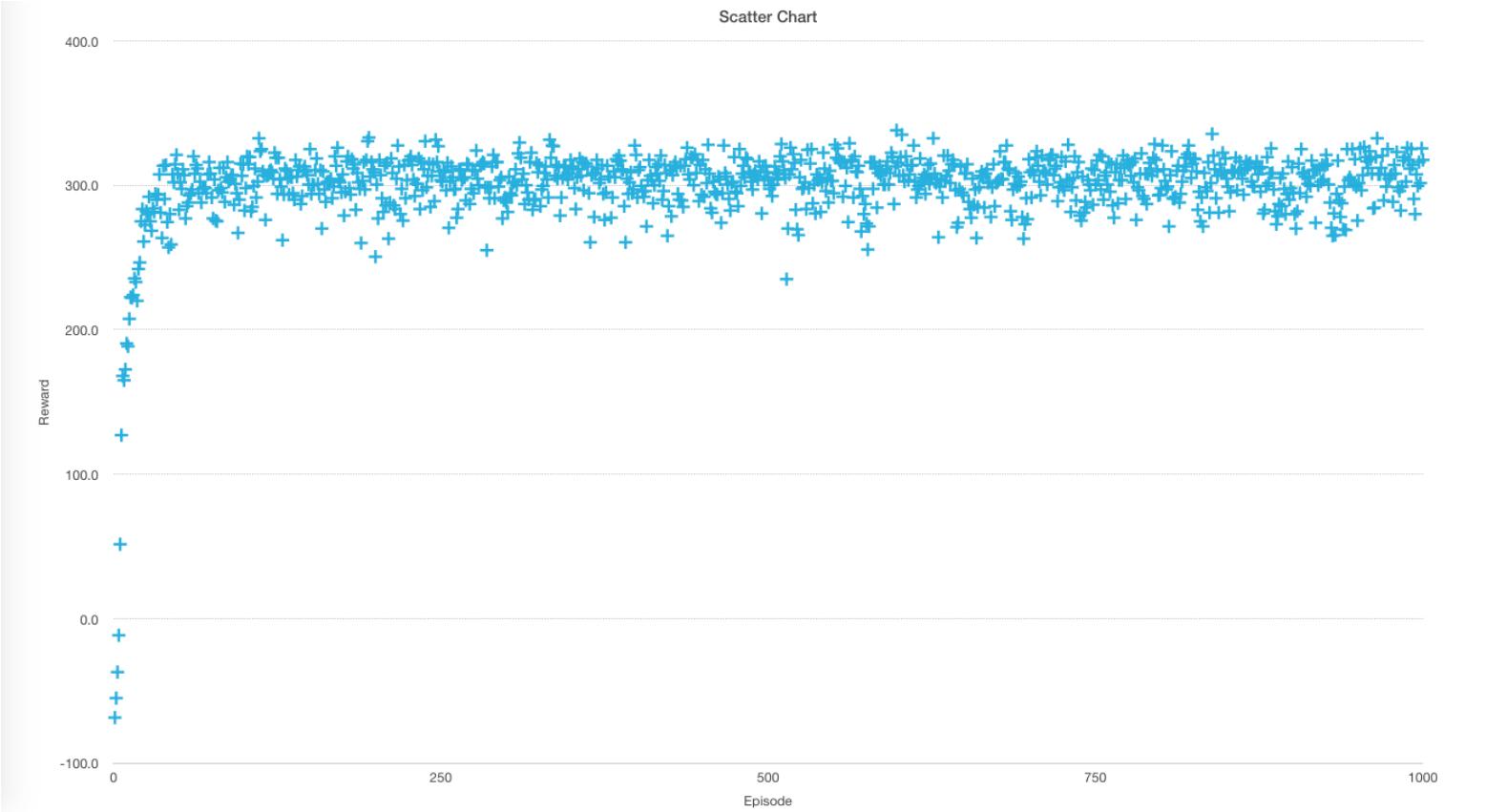
b) Because the agent knows the location of the thief, it performs a lot better. After learning for the first 10 episodes, the agent was able to consistently score rewards

higher than 100. The fact that the agent knew the location of the thief it can make decisions and update its future policies based on this knowledge.

c) The best learning rate is: 0.33 and the best epsilon is: 0.02. I feel like this is the best options because when running the simulation, the rewards were increasing at a faster rate and early on in the episodes. This means that the agent learns at a better rate and results in a higher performance. The redress started to hit positive at around episode 5, greater than 100 at around episode 6, greater than 200 at around episode 15, and greater than 300 at around episode 50. This shows that the rewards have been consistently going up, even though early on there definitely is a slight fluctuation, which is expected knowing that the robot is learning about its environment.

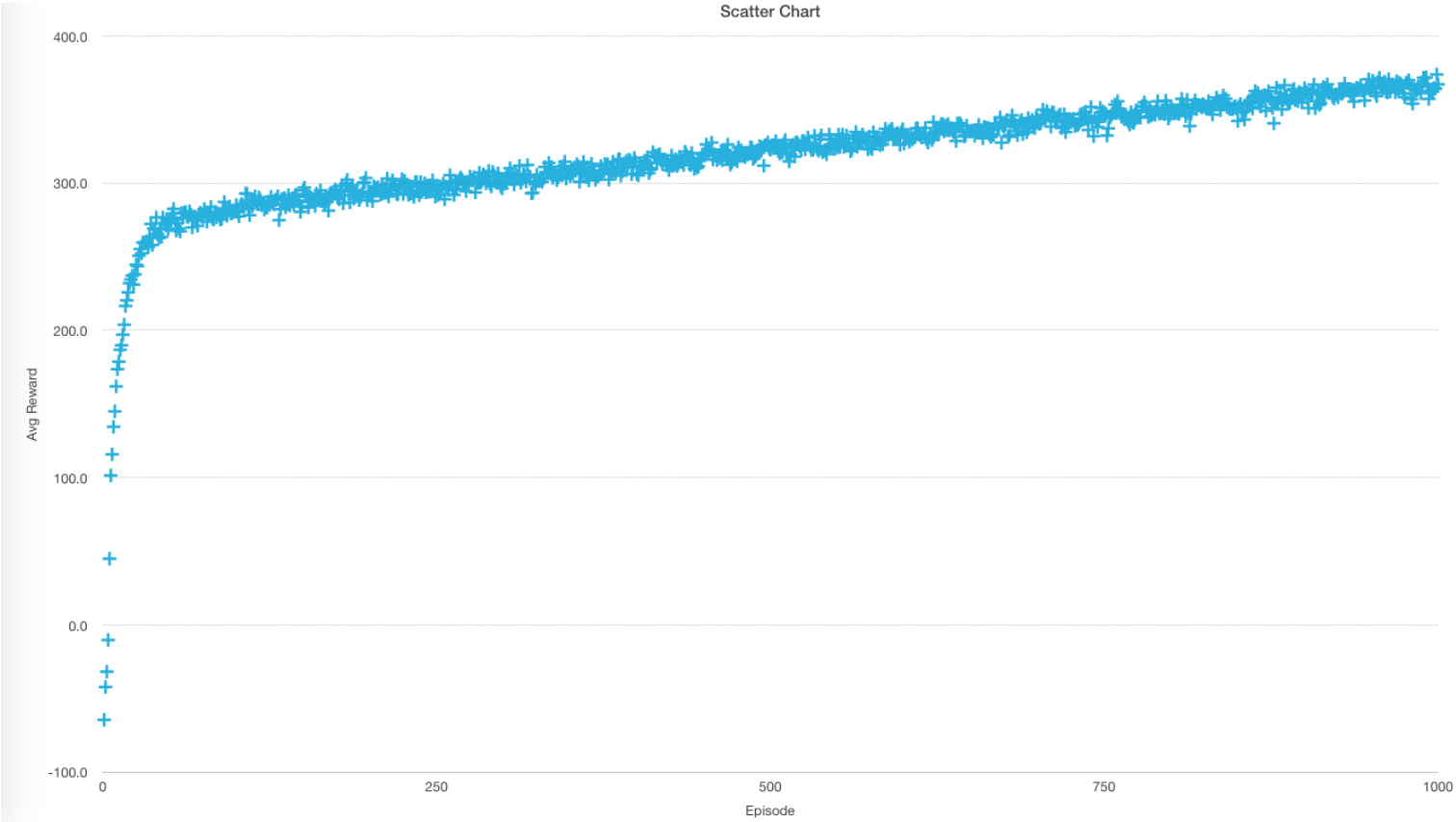
---

3) Episodes vs Expected Discount:



I notice that initially the agent learns at a quicker rate until it hits its peak. This peak, as we can observe from the plot fluctuates around a reward of 300. At this point the average reward keeps at 300 without much change overall.

Average of 9 Episodes vs Expected Discount:



Observing the averages this time, we get the same result with respect to the fact that the agent learns greatly initially. The plots are jumping greatly until the rewards get to around 280 now. However, instead of fluctuation around the 300 reward mark, we see that the plot is constantly increasing. Even getting an average of past 300, which proves to be better than just observing the rewards from a single simulation.