Matthew Hoffman
mfhoffm2
CS 440 MP1 Part 2

1) a) The first episode has the agent getting a negative reward, presumably making a guess at the start and wandering onto the ice. Then, after it learns that every other episode has a reward of 0.0. The Agent appears to have learned to just go straight up into the top wall and sit there, with nothing happening.
1) b) The first ~20 rounds all had negative rewards as the agent explored and failed to find the end goal. The negative reward started out bad, around -10, but quickly improved as it found out how to avoid the dangerous areas, presumably finding feasible policy that didn't have any losses, but deviating every now and then to explore and get a small loss. Then eventually on round 20 it got a reward of 218 and appears to have found the end goal and how to get an actual reward. After this, it would have a policy leading it to that high reward, which shows in that it continued to get high rewards, now hovering in the 90-200 range. The variance comes from it abandoning its known good policy and searching elsewhere (and not finding anything of value).
1) c) Each round is pretty consistently a negative reward, and a pretty serious one at that. Usually in the -10 to -50 range. Even if a good policy could be found, it would be deviated from so often that it wouldn't matter much. This nearly random system gives random results, and there are many more negative rewards to randomly stumble onto than positive ones.

2) a) With the thief running around invisibly, the agent never actually was able to randomly stumble around and find the goal. It had negative results for essentially every episode, on the order of -10. The thief kept disincentivising the correct path, and the path was long enough that the agent wouldn't randomly find its way there frequently enough to set up a policy path from the start to the end.

2) b) There are some initially bad episodes, with large negative rewards when the robot randomly wanders into the thief. However, immediately these negative rewards start to decrease in severity as the robot learns to avoid the thief in its policy and mostly hit it due to the epsilon randomness. Eventually, the reward is found after around 8 episodes and the total reward shoots up into the high 200s and stays there for the rest of the episodes. The robot has found a policy to the reward that lets it actively avoid the thief moving around, and can consistently get the reward.

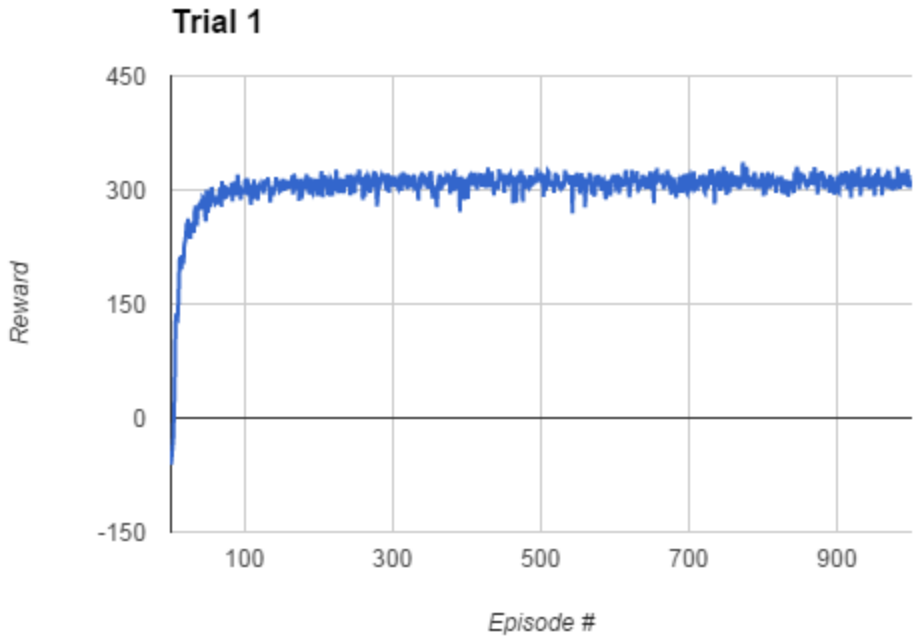
2) c) I first started fiddling with the epsilon value. The first thing that I noticed was that increasing the value expectedly decreased the number of episodes required for the machine to find the reward and settle into a pattern of success. The base value of 0.05 took around 8, while 0.1 took only 6 or so. 0.075 took very few as well, but also had the interesting result of quickly finding a suboptimal path getting around 40 for a final reward for a while before eventually exploring and finding a more optimal one and getting ~250 as a reward. There was also a tradeoff between convergence speed and result as higher epsilon values would have slightly less average final reward as it continued to explore unnecessarily into negative rewards. This trend continued with lower epsilon values as well, with values like 0.025 regularly getting rewards just over 300 but taking around 100 episodes to reach that point. (It reached positive rewards much before that, having a quick initial increase turning into a slower steady one until it stabilized eventually). This breaks down above around 0.1 and below about 0.001. Too high and the randomness prevents a

good policy from being found and there are just negative results. Too low and the increase is too slow and policy is still slowly improving at a relatively low value even after 1000 episodes. 0.025 seems like a good middle ground, converging relatively quickly to a rather good average result.

A lower learning rate definitely slows the speed at which the results improve. Values less than 0.125 seem to be mostly just slower. Some of the values on the high end fluctuate a little more maybe, but are still consistently good. Above this though, as early as 0.15, improvement actually slows down. The policy changes too quickly and supports too many bad initial leads.

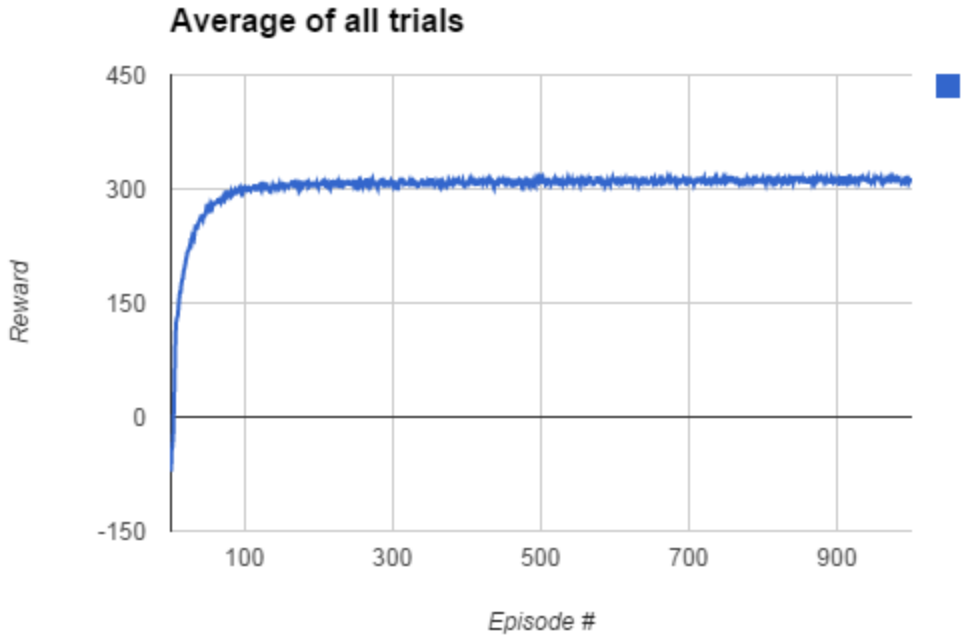
I believe the best values are around 0.025 for epsilon and 0.125 for the learning rate.

3)



I first notice there seems to be some sort of logarithmic convergence to a value (in this case about 300), however there is also a good deal of noise making the graph line rather “fuzzy”. A lower epsilon might reduce this noise, but would make the graph take longer to reach its peak.

Matthew Hoffman
mfhoffm2
CS 440 MP1 Part 2



This graph looks a lot like the other one, except most of the noise is averaged out and reduced. This means that across the trials, the reward seemed to improve in a pattern, each time increasing somewhat similar amounts. The noise affected any given episode's reward, but not as much the overall convergence. Also, since the noise mostly averaged away to a flatter line, that means it likely is "random" and not really a part of a larger pattern, just noise in the result.