Samuel Traylor, Vaughn Bangash

Professor Mattei

CMPS 3140

May 4, 2021

Project 5

1. To get the optimal policy to cross the bridge, all I needed to do was set the noise to 0.0 and the discount relatively high (I tried 0.9) which worked because it allowed the policy not to worry about falling into the pit. You could also have tried increasing the discount rate to increase the urgency of it going to the terminal state, however in practice this would likely cause the agent to minimize losses by jumping into the pit quickly.

3. For policy 3a, the policy needed to prefer the closer exit, which is why I set a large negative reward for living (-2.0) and a discount of 0.5, which created urgency for reaching a terminal state. To get it to risk the pits I again lowered the noise to 0.0. The agent correctly made a B-line to its goal as well as preferring the closer exit. For policy 3b, I kept the discount the same, but I raised the answer noise to 0.2 so there would be *some* threat of falling in the pit by accident, but I also made sure the living reward was still punishing enough at –1.0 so that the policy would pick the close exit. In 3c, I had to use an answer discount of 1.0, effectively eliminating the "diminishing returns" aspect so that the larger score exit would be preferred even if it was further. To get it to take the risky route I again set noise to 0.0, and in order to make doubly sure it took the time to reach the further exit, I set a much less significant living reward of –0.05. For 3d, I kept the discount and living reward the same so that the distant exit would be preferred, but changed the noise back to 0.2 so that the risk of falling into the pit by accident

reappeared. For 3e, to keep the agent wandering around without choosing a terminal state, the policy needed to have no discount (so the rewards would be just as good later as they are now) as well as having a noise of 0.5 to strongly discourage risking the pits, as they would end the infinite loop. Last and most importantly, setting the living reward to a miniscule but *positive* 0.01, meaning there's an ever-increasing reward for not leaving or falling in the pit.

7. The epsilon value, the fraction of the time that the agent chooses a random action, is essentially a value that controls the agent's inclination towards exploration vs exploitation. Increasing it early means the crawler will explore more via a higher chance of taking a random action at any timestep. Keeping it high after that however risks wasting time exploring when the best movements are already learned, in which case you waste time because even though you've already successfully crawled forward its looking for new, even better options. Conversely, a low learning rate, especially coupled with low incentive for exploration (low epsilon) could mean the baby steps taken by the crawler are encouraged because they have a proven reward as opposed to exploring and finding that outstretching the crawlers arm more allows for more distance.

8. This problem actually can't be solved: it's impossible to get the optimal policy 99% of the time in 50 episodes, because even a learning rate of 1.0 doesn't allow for that success rate in 50 or less episodes regardless of epsilon value.

9. Pacman is doing far better than he had in any other previous algorithm/agent type. On larger mazes however, he does terrible, and changing the learning rate does not help him. This is because each board configuration is its own state with its own Q-values. There is no way to generalize that running into a ghost is bad for all positions, making the kind of connections that can be made on gridworld, for example.