Sebastian Crowell

Assignment 2 attempt

April 24, Late

COMP 560

Reinforcement Learning (Exploration/Exploitation)

For the assignment, I failed to actively have a program that could complete a round of golf from state to state, as I lack the necessary background to discern how the floating value needs to be kept in scope of the code itself.

Instead I will describe what I was trying to do and observe for the answers to the problems. My goal was to set the strings in the text file as parts separated by the backslashes. By doing so I could check the input lines for if the text was contained before the slash. If it was found, I added it to a random pool of lines, selected one, went to the text before the next backslash, and wanted to recur this process. The recursion failed because of the scope of the c program couldn't maintain the address between the layers of functions and the variables began to become more difficult to control. Somehow, the c code that should have works, that I had checked, couldn't maintain the decimal point values at the end of the recursion and the value was dropped off, so I couldn't maintain reinforcement learning equations and had to observe everything manually. I had to give things that had a higher percentage of completing while also getting the closest to the hole the best value. This made everything in a way, a twisted combo of exploration and exploitation, as I would choose a random value to apply it never had the correct reward and it always was selecting the highest value next jump.

For answering the questions, number one related to changing the exploration value from values close to 0 or 1. Starting it close to 1 should have made the program iterate less as it expected to be close to the end of the golf game and should have made the regret affect the program more heavily. Starting close

to 0 should make rewards provide more value and should have made more rewards as almost anything could be in an increase in position, more iterations on average. Number two was sort of the main issue and why the program stops at 10 no matter what at the moment. If I don't terminate it early it will run for hours and never complete a game of golf as the values bounce from extremely small values that barely change (not visible outside of the ide's interface for inspection). Had the program worked correctly, I would have chosen to multiply the value being used by exploration by 3 * the total number of lines in the input file, this would have provided sufficient results as I could force the program to have an improving transition. Number 3 the discount value affects the total amount of iterations before the program has to terminate, as we can determine if higher reward now is better than later. Ideally, this would be what controls the answer to number 2, so that we have a good representation of the optimal policy. It would probably be a good mixture being close to .5 to make sure not to shoot in a way that would cause an infinite amount of repetition. Number 4 discusses the epsilon value for the program as it chooses where to crawl. After trying for a long time and working through the calculations, the epsilon values that make the most sense were between 7% and 13% depending on how much randomness is wanted.

Spending a lot of time on this project wasn't ideal, as I was fairly sure the calculations were not hard. That was correct, but the problems came from my inexperience with c coding. I had to try this assignment by myself to see where I'm at.