Lab 12 Analysis

Lab: Reinforcement Learning
CSC 261
File: Analysis.pdf
Summary: Analysis of our td and q-learning algorithms

Part A: Temporal Difference Learning

Test

We tested the program using our implementation of PASSIVE-TD-AGENT using the pseudocode provided in AIMA Figure 21.4 (pg. 837). We used one of our implementation of policy iteration as well to get the optimal policy for each state. The values for gamma and epsilon were 0.99999 and 0.0001 respectively. We tried to match the exact policies mentioned in AIMA by highly valuing the future states as well as keeping the error margin as low as possible. We ran about 10,000,000 trials to get the utility values as close to the number provided in AIMA Figure 21.1 (b), although there was one state where the utility value differed in thousand decimal place. An interesting observation that we made was that there was a non-terminal state whose utility was 0.0. In other words, its utility was never updated.

Predict

Since the 16x4 world is bigger than the 4x3 world, we predict that the number of trials required to get the exact utility values for each state will be higher. Given that the 4x3 world had 10,000,000 trials, the number of trials required for the TD learner would be larger than that, approximately 4.5 times more, which is proportional to the number of states in each world. We want to make sure that we cover all possibilities before the utilities for each state are finalized.

Experiment

For the experiment part, we used the same parameters as the 4x3 world. We conducted as many trials as we did for the 4x3 world and the numbers that it produced were very close to the ones generated by value iteration, although there were some utilities that were off in the thousands decimal place. This could not be improved even by increasing the number of trials. In fact, we got a lot of fluctuations in numbers as we increased the numbers of trials. Additionally, similar to the 4x3 world, there were a couple of states whose utility was never updated (see Figure 1).

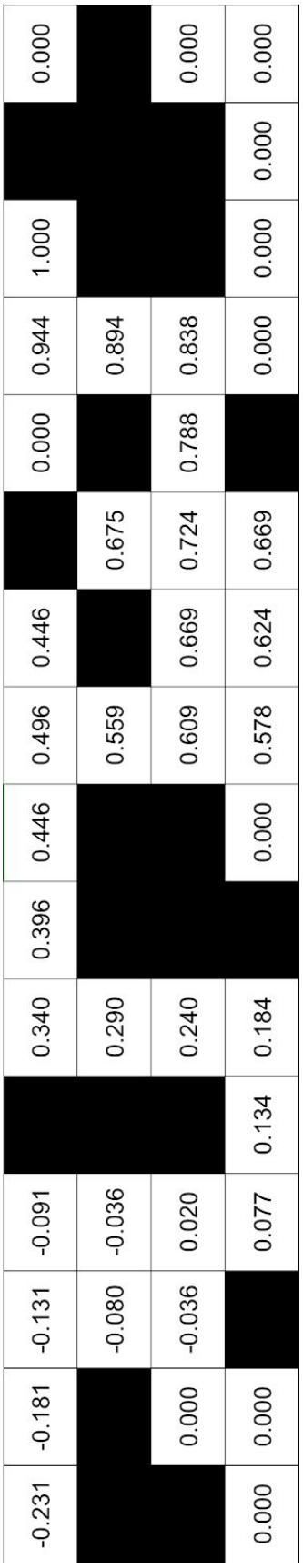

Figure 1: The utility value for each state in the 16x4 world as determined by the TD learner.

Reflect
Contrary to our assumption, there isn't any significant difference between the number of trials between 4x3 world and 16x4 world. This was a bit surprising. We have a couple of theories that could explain this. Since the learning curve flattens relatively quickly compared to when it was rising, we might have run more trials than needed for 4x3 world. Hence, we could have probably used less number of trials for the smaller world and still get an accurate representation of the utilities for each state. Another theory is that since trials are one entire trip to a terminal state, each trial is longer, maybe by enough to make up the difference in difficulty learning. Hence, there should be no discrepancy in the number of trials between the two worlds. Moreover, with regards to the 0.0 utilities, the agent with the policies could never reach those states. Hence, their utility values were never updated.

Part B: Q-Learning
Test
We tested the program using our implementation for this part as well. We chose to do 1,000,000 trials, as it ran fairly quickly but seemed to be enough to get real results. Some trial and error showed that, in order to get the optimal policy, the reward parameter had to be 1, and we needed to be willing to explore 200,000 times. We would expect there to be some trade off here, but it seems hard to get either of these much lower. Interestingly, the utilities have a fair bit of error at this level, but it doesn't compromise the policy. One thing to note in this analysis: we had the agent take the last action that had maximal exploration value. Doing the same thing but taking the first action with maximal exploration value does not yield the optimal policy with these parameters. That we needed a reward of 1 was what we expected, because it meant that we would always choose to explore over any other action, as all actions had utilities < 1.

Predict
We think it will be a little harder, but maybe not that much harder to learn the optimal policy for the 16x4 world. We will start by trying the same parameters but anticipate possibly needing the up the number of trials. It also seems possible to us that the number of trials and attempts might not need to be increased, just because each trial is longer in the 16x4 world, and so we will learn the state-action values in less trials.

Experiment
We ran it with the same parameters as in the 4x3 world, and there were 5 states in which the policy was erroneous. After trying to increase the values, there were some states that we could not get to q-learner to learn about even for quite high values of the parameters, though. In an attempt to mitigate this, we changed qlearn from taking the last action to taking a random action among the "best" actions - we thought maybe it was systematically avoiding certain states, and this might help remedy it. This did not improve it, so that idea's out. Even running for a very large number of trials (20,000,000) with many required attempts (1,000,000) did nothing to teach the q-learner about the states that it is not learning about, and this took something like five minutes to run. We also tried lowering the value of maximum reward - it doesn't seem obvious why this would help, as it would just cause it to explore less, but it seemed worth a try. This did not improve it either. The changes we're making don't seem to be getting us closer to the optimal policy on these states that the

system is finding really hard to learn about - these states are state 7, 36, 40, 50, and 51. Changing the parameters let us change which states were wrong, but it was always a tradeoff, rather than an improvement.The general thing that seems to be similar about these states are their proximity to -1 terminal states, but we're not sure why this is relevant.

The policy recommended by our q-learner:

Reflect

The results here were clearly different than what we expected. In particular, we couldn't get the optimal policy from our q-learner, even with a fair bit of fiddling. Interestingly, using the parameters that worked for the 4x3 world was actually the best we could do on the 16x4 world as well. Our best guess is that the q-learner is very rarely entering some states, which is making it not evaluate them accurately - upping the number of trials and required attempts by manyfold did not fix the problem, though, so it's a little hard to interpret. Because the q-learner was getting very accurate utilities for some states and inaccurate utilities for a couple in the 4x3 world, it seems we may be having a similar problem.