Alex Anderson

EECS 658

Assignment 8

8 December 2022

Question Answers

Question 1:

First, select a random state from the available states = (0 to 24). For each episode record the sequence of states and rewards until it reaches the terminal state. Define the G(s), the discounted reward from the selected random state to the terminal state. Now we create a value matrix V(s) for each episode. We keep track of the cumulative number of visits and cumulative total discounted reward G(s). In the Monte-Carlo first-visit algorithm, matrices N(s) and S(s) are persistent across episodes These are calculated the first time a state is entered, calculated below:

Increment counter of visits: $N(s) = N(s) + 1$

Increment total return $S(s) = S(s) + G(s)$

At the end of all the episodes, we will estimate the value matrix:

$V(s) = S(s) / N(s)$

Finally, go through episodes until all the states are represented in the value matrix. This method was selected because the steps are outlined in the lecture.

Question 2:

We follow the same convergence method except for the fact that for Monte-Carlo every visit we compute the matrices every time we enter that state each episode using the formulas below:

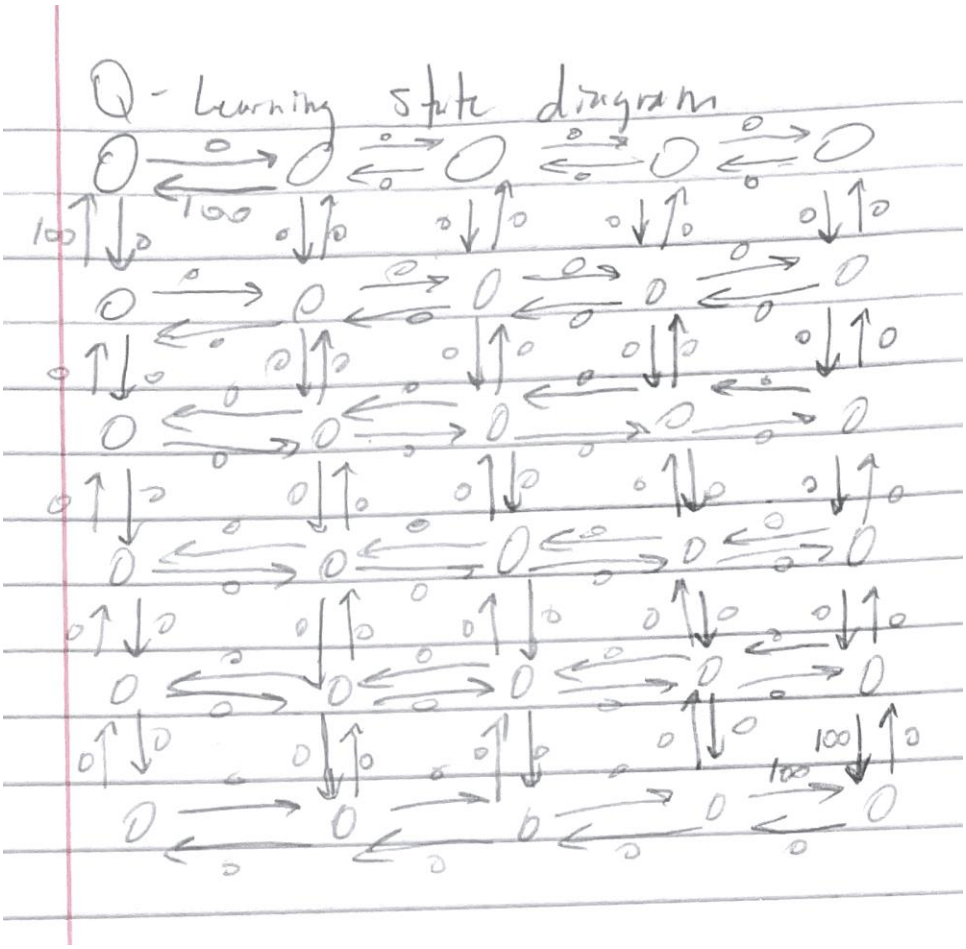
Increment counter of visits: $N(s) = N(s) + 1$

Increment total return $S(s) = S(s) + G(s)$

Part-1 and Part-2 converged in the same number of episodes.

Question 3:

Q-Learning state diagram



Question 4:

The conference method I used for the Q-Learning algorithm is as follows, first we set the gamma = 0.9 and define the rewards matrix. Then initialize the Q-matrix and run a constant number of iterations (say > 500) as we have only 25 states for each episode. Next, select a random initial state and perform the states below until we reach the terminal state. Then randomly select one among all possible actions for the current state. Use this possible action and go to the new state. Finally get the maximum Q-value for this next state based on all possible actions and update the Q-value matrix for the selected state. The algorithm converges after 100 iterations.

Question 5: Below Optimal policies were followed for state = 7

State 7 ⟶ State 2 ⟶ State 1 ⟶ State 0 (terminal state)

State 7 ⟶ State 6 ⟶ State 1 ⟶ State0 (terminal state)

State 7 ⟶ State 6 ⟶ State 5 ⟶ State0 (terminal state)

Question 6:

First, similarly to part 3, we set the gamma = 0.9 and define the rewards matrix and initialize the Q-matrix. Next, we run a constant number of iterations (say >= 100) as we have only 25 states for each episode. Then, select a random initial state and perform the below states until we reach the terminal state. Select the action with highest value in the Q-matrix. Use this possible action and go to the new state. Finally, get the maximum Q-value for this next state based on all possible actions and update the Q-value matrix for the selected state.

Question 7:

State 7 ⟶ State 2 ⟶ State 1 ⟶ State 0 (terminal state)

Question 8:

SARSA converged faster than Q-learning most likely because it takes a greedy approach by traversing less paths.

Question 9 – Question 12:

I did not get to this portion of the assignment.