Homework 3

Akshay Sharma (akshaysh)

Answer 1

a)

Adjacency matrix:

$$A = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \end{bmatrix}$$

b)

Given,

$$X = \begin{bmatrix} 1 & -1 \\ -2 & 0.5 \\ 1 & 3 \\ 0 & -1 \end{bmatrix}$$

$$AX = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & -1 \\ -2 & 0.5 \\ 1 & 3 \\ 0 & -1 \end{bmatrix} = \begin{bmatrix} -1 & -0.5 \\ -1 & 2.5 \\ -1 & 3.5 \\ 2 & 1 \end{bmatrix}$$

This means,

$$X^{'} = \sigma(AX) = \begin{bmatrix} 0 & 0 \\ 0 & 2.5 \\ 0 & 3.5 \\ 2 & 1 \end{bmatrix}$$

c)

The Degree matrix,

$$D = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 3 \end{bmatrix}$$

$$D^{-1} = \begin{bmatrix} 1/2 & 0 & 0 & 0 \\ 0 & 1/3 & 0 & 0 \\ 0 & 0 & 1/2 & 0 \\ 0 & 0 & 0 & 1/3 \end{bmatrix}$$

$$A^{'} = D^{-1}A = \begin{bmatrix} 1/2 & 1/2 & 0 & 0 \\ 0 & 1/3 & 1/3 & 1/3 \\ 0 & 1/2 & 1/2 & 0 \\ 1/3 & 0 & 1/3 & 1/3 \end{bmatrix}$$

Now based on $A^{'}$, we will calculate the new features $X^{'}$,

$$A^{'}X = \begin{bmatrix} 1/2 & 1/2 & 0 & 0 \\ 0 & 1/3 & 1/3 & 1/3 \\ 0 & 1/2 & 1/2 & 0 \\ 1/3 & 0 & 1/3 & 1/3 \end{bmatrix} \begin{bmatrix} 1 & -1 \\ -2 & 0.5 \\ 1 & 3 \\ 0 & -1 \end{bmatrix} = \begin{bmatrix} -0.5 & -0.25 \\ -1/3 & 0.833 \\ -0.5 & 1.75 \\ 0.66 & 0.33 \end{bmatrix}$$

This means,

$$X^{'} = \sigma(A^{'}X) = \begin{bmatrix} 0 & 0 \\ 0 & 0.833 \\ 0 & 1.75 \\ 0.66 & 0.33 \end{bmatrix}$$

2

a)

Expected immediate reward on taking "Action A" in "State 3"

$$\mathbb{E}[R(S_3, Action\ A)] = 0.25 * r(S_1) + 0.75 * r(S_2) = 0.25 * 6 + 0.75 * 2 = 3$$

Expected immediate reward on taking "Action B" in "State 3"

$$\mathbb{E}[R(S_3, Action\ B)] = 0.5 * r(S_4) + 0.5 * r(S_5) = 0.25 * 3 + 0.75 * 1 = 2$$

So, "Action A" give a higher expected reward in "State 3"

b)

If "Action A" is taken in "State 3", we get the following system of linear equations:

$$\begin{bmatrix} S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \end{bmatrix} = \begin{bmatrix} 6 \\ 2 \\ 0 \\ 3 \\ 1 \end{bmatrix} + \gamma \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0.25 & 0.75 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.5 & 0.5 \\ 0 & 0 & 0 & 0.5 & 0.5 \end{bmatrix} \begin{bmatrix} S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \end{bmatrix}$$

Solving the system of linear equations we get,

$$S_1 = 6.0 + \gamma S_3$$

$$S_2 = 2.0 + \gamma S_3$$

and,

$$S_3 = \gamma(\frac{S_1}{4} + \frac{3S_2}{4})$$

This gives us,

$$S_3 = \frac{3\gamma}{1 - \gamma^2}$$

So for $\gamma = 0.8$,

$$S_3 = 6.66$$

If "Action B" is taken in "State 3", we get the following system of linear equations:

$$\begin{bmatrix} S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \end{bmatrix} = \begin{bmatrix} 6 \\ 2 \\ 0 \\ 3 \\ 1 \end{bmatrix} + \gamma \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0.5 & 0.5 \\ 0 & 0 & 0 & 0.5 & 0.5 \\ 0 & 0 & 0 & 0.5 & 0.5 \end{bmatrix} \begin{bmatrix} S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \end{bmatrix}$$

Solving the system of linear equations we get,

$$S_3 = \gamma \frac{S_4 + S_5}{2}$$

$$S_4 = 3.0 + \gamma \frac{S_4 + S_5}{2}$$

$$S_5 = 1.0 + \gamma \frac{S_4 + S_5}{2}$$

This gives us,

$$S_4 = \frac{3 - \gamma}{1 - \gamma}$$

$$S_5 = \frac{1 + \gamma}{1 - \gamma}$$

and,

$$S_3 = \frac{2\gamma}{1 - \gamma}$$

So for $\gamma = 0.8$,

$$S_3 = 8.0$$

This means the expected discounted future reward for "Action B" is higher if $\gamma = 0.8$.

c)

For the future expected reward of "Action A" to be higher than that of "Action B"

$$\frac{3\gamma}{1 - \gamma^2} > \frac{2\gamma}{1 - \gamma}$$

$$\frac{3 - 2(1 + \gamma)}{1 - \gamma^2} > 0$$

Then for $\gamma \neq \pm 1$,

$$(1 + \gamma) < 1.5$$

$$\gamma < 0.5$$

Answer 3

a)

At the start, $Q(S_1, A) = Q(S_1, B) = 0$, so we take action A,

$$Q(S_1, A)_1 = Q(S_1, A)_0 + 1.0(1.0 + 0.5 * \max_a(Q(S_2, a)_0) - Q(S_1, A)_0)$$

Now since S_2 has only one action, "right"

$$Q(S_1, A)_1 = 0 + (1 + 0.5 * 0 - 0) = 1.0$$

As action "B" was not selected, its Q value will stay the same,

$$Q(S_1, B)_1 = 0$$

b)

As all the states $S_i, i \neq 1$, have only one action "right", so the state S_1 will not visited again after till the N^{th} step. Now as $N > 1000$ this means $5 < N$, so

$$Q(S_1, A)_5 = Q(S_1, A)_1 = 1.0$$

$$Q(S_1, B)_5 = Q(S_1, B)_1 = 0$$

This is similar for all the states such that any state $S_i, i \epsilon [N], i \neq 0$ will get visited after every $N + (i-1)^t h$ step, and their Q-values be updated after $(N+i)^t h$ state.

c)

As stated in the previous part, state S_1 will be visited after the N^{th} step.
Now $Q(S_1, A)_N = Q(S_1, A)_1 = 1.0$, and $Q(S_1, B)_N = Q(S_1, B)_1 = 0$, so action A will be chosen again at the $(N+1)^{th}$ step, and thus the Q values will be,

$$Q(S_1, A)_{N+1} = Q(S_1, A)_N + 1.0(1.0 + 0.5 * \max_a(Q(S_2, a)_N) - Q(S_1, A)_N)$$

Now since any state S_1, will get visited after every $N + 1^{th} step$, $Q(S_2, a)_N = Q(S_2, a)_1$.

$$Q(S_2, right)_N = Q(S_2, right)_2 = Q(S_2, right)_1 + 1.0(1.0 + 0.5 * \max_a(Q(S_3, a)_1) - Q(S_2, right)_1)$$

$$Q(S_2, right)_N = Q(S_2, right)_2 = 0 + (1 + 0.5 * 0 - 0) = 1.0$$

$$Q(S_1, A)_{N+1} = 1.0 + (1.0 + 0.5 * 1.0 - 1.0) = 1.5$$

and

$$Q(S_1, B)_{N+1} = 0.0$$

Now,

$$Q(S_1, A)_{N+5} = Q(S_1, A)_{N+1} = 1.5$$

and

$$Q(S_1, B)_{N+5} = Q(S_1, B)_{N+1} = 0.0$$

d)

From the previous part, we can see that action B will never get taken, and hence

$$Q(S_1, B)_\infty = 0.0$$

Here ∞ means at convergence.
For $Q(S_1, A)$, we can see this value will get updated every $(N+1)^{th}$ step, using the equation,

$$Q(S_1, A)_i = Q(S_1, A)_{i-1} + 1.0(1.0 + 0.5 * \max_a(Q(S_2, a)_{i-1}) - Q(S_1, A)_{i-1})$$

where $i\%N = 1$. Also S_2 only has one action "right". For all such i, $Q(S_2, right)_{i-1} = Q(S_1, A)_{i-1}$
Thus we can write,

$$Q(S_1, A)_i = Q(S_1, A)_{i-1} + (1.0 - 0.5 * Q(S_1, A)_{i-1})$$

$$Q(S_1, A)_i = 1.0 + 0.5 * Q(S_1, A)_{i-1} = \sum_{j=0}^{i} (0.5)^j$$

This means,

$$Q(S_1, A)_\infty = \frac{1.0}{1 - 0.5} = 2.0$$

So at convergence,

$$Q(S_1, A) = 2.0$$

and

$$Q(S_1, B) = 0$$

Answer 4

a)

On-policy methods use the data generated by the current policy to update the policy, thus they require the ability to interact with the environment to generate samples according to the current policy. eg: PPO, TRPO.
Off-policy methods can use data generated by any policy to improve the current policy. This means that they can use previously stored data for the updates. This also makes them sample efficient as the same samples can be used for multiple updates. eg: DQN, DDPG.

b)

When we train a DQN model, we break the assumption of i.i.d data which is usually a requirement for training deep networks. This happens because the actions taken in any step, affect the next states which are visited in that episode. So to break this interdependence we use an experience replay, from which we can randomly sample non consecutive data which will not be correlated to each other by a huge amount.

c)

Q-Learning tries to learn the optimal action-value function at each state, which can be later used to give the optimal policy, whereas policy gradient methods use the policy itself for maximize the returns. As the policy gradients directly try to optimize the actual objective (getting the optimal policy), they tend to perform better than Q-Learning.

d)

The critic provides estimates of the value function which are used to estimate the current value of the state. If a critic is not used, we will have to run a full episode till it reaches a terminal state, to get the complete discounted reward, but by estimating the discounted rewards at any step, we can truncate the episode at any arbitrary length, and hence can be more efficient.