Sumeet Shanbhag

Student ID : 641020714

Reinforcement Learning Assignment #5

1. **The first episode of an agent interacting with an environment under policy π is as follows:**

Timestep	Reward	State	Action
0	--	X	U1
1	16	X	U2
2	12	X	U1
3	24	X	U1
4	16	T	

Given: discount factor, γ=0.5, step size α = 0.1 and $q\pi$ is initially zero. The general function for calculating Action Value for 2 Step SARSA:

$$Q(S0,A0) = Q(S0,A0) + \alpha(R1 + \gamma R2 + \gamma 2\ Q(S2,A2) - Q(S0,A0))$$

TimeStep1: Q(X,U1) = 0 + 0.1(16 + 0.5(12) + 0.52(0) – 0) = 2.2

TimeStep2: Q(X,U2) = 0 + 0.1(12 + 0.5(24) + 0.52(2.2) – 0) = 2.455

TimeStep3: Q(X,U1) = 2.2 + 0.1(24 + 0.5(16) + 0.52(0) – 2.2) = 5.18

TimeStep4: Q(X,U1) = 5.18 + 0.1(16 + 0.5(0) + 0.52(0) – 5.18) = 6.262

From the calculations done above, $\boldsymbol{q_\pi\ (X,U1) = 6.262}$, $\boldsymbol{q_\pi\ (X,U2) = 2.455}$.

2. **What is the purpose of introducing Control Variates in per-decision importance sampling?**

In per-decision importance sampling, due to the importance sampling ratio,

$$\rho = (\pi(Ak|Sk) / b(Ak|Sk))$$

However, just applying this ratio directly throughout the trajectory introduces a significant problem: high variance in our estimates. This issue is further exacerbated if our behavior policy takes an action that our target policy would never take, leading to ρ=0 and rendering our return estimate G zero.

To counteract this variance, we introduce a Control Variate. A control variate is a correlated variable that, when used correctly, can reduce the variance of our estimator

without introducing bias. The idea is to adjust our original estimator by subtracting the expected value of the control variate.

The inclusion of the control variate in the estimation can be represented as:

$$G_{t:h} = \rho_t(R_{t+1} + \gamma G_{t+1:h}) + (1 - \rho t)V_{h-1}(S_t)$$

Where the term (1 - ρt)Vh-1(St) is the control variate. This component ensures that even if our importance sampling ratio becomes zero at some point, our target estimate remains meaningful, and isn't entirely reduced to zero. When the importance sampling ratio is one, the control variate has no effect on the expected update, thus preserving the unbiasedness of the estimate.

3. **In off-policy learning, what are the pros and cons of the Tree-Backup algorithm versus off-policy SARSA (comment on the complexity, exploration, variance, and bias, and others)?**

- **Complexity:**

  Tree-Backup: Tree-Backup operates by considering all actions at every step, rather than just the action taken by the behavior policy. This leads to an expanded tree-like backup diagram. The computational complexity is higher because it requires considering the value of each action at each step.

  Off-policy SARSA: Off-policy SARSA is a bit simpler in its backup diagram. It uses the importance sampling ratio to correct the mismatch between the target and behavior policy. The computational complexity is typically lower than Tree-Backup because it considers only the action taken by the behavior policy.

- **Exploration:**

  Tree-Backup: Tree-Backup doesn’t rely on importance sampling, which can sometimes have extreme values. This allows the algorithm to explore more safely. Because it considers all actions, it can gather more comprehensive information about the environment.

  Off-policy SARSA: Its exploration is influenced by the behavior policy. If the behavior policy is more exploratory, off-policy SARSA will gather more diverse data. However, if the importance sampling ratio is extreme, it can affect the updates and thereby exploration.

- **Variance:**

  Tree-Backup: Tree-Backup generally has lower variance than methods that rely on importance sampling because it doesn't use importance sampling.

  Off-policy SARSA: Importance sampling can introduce high variance, especially when the behavior policy is far from the target policy. This can make learning less stable.

- **Bias:**

Tree-Backup: It might have a slight bias because it doesn't use importance sampling to correct the difference between the target and behavior policies. Instead, it uses expected values over all actions.

Off-policy SARSA: Uses importance sampling to correct the bias introduced by the mismatch between the target and behavior policies. However, if the behavior policy has a zero probability for actions taken by the target policy, then the importance sampling ratio is undefined, which can introduce bias.

4. Exercise 7.4 of the textbook (page 148).

We have:

$$G_{t:t+n} = Q_{t-1}(S_t, A_t) + \sum_{k=t}^{\min(t+n,T)-1} \gamma^{k-t} \left[R_{k+1} + \gamma Q_k(S_{k+1}, A_{k+1}) - Q_{k-1}(S_k, A_k) \right].$$

Expanding the above equation:

$= Q_{t-1}(S_t, A_t) + R_{t+1} + \gamma Q_t(S_{t+1}, A_{t+1}) - Q_{t-1}(S_t, A_t) + \gamma R_{t+2}, + \gamma^2 Q_{t+1}(S_{t+2}, A_{t+2}) - \gamma Q_t(S_{t+1}, A_{t+1}) + \gamma^2 R_{t+3}, + \gamma^3 Q_{t+2}(S_{t+3}, A_{t+3}) - \gamma^2 Q_{t+1}(S_{t+2}, A_{t+2}) + \ldots.. + \gamma^{n-1} R_{t+n}, + \gamma^n Q_{t+n-1}(S_{t+n}, A_{t+n}) - \gamma^{n-1} Q_{t+n}(S_{t+n-1}, A_{t+n-1})$

Upon cancelling the similar terms, we get:
$= R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots. + \gamma^{n-1} R_{t+n} + \gamma^n Q_{t+n-1}(S_{t+n}, A_{t+n})$

This is equivalent to the n-Step returns for SARSA:

$$G_{t:t+n} \doteq R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{n-1} R_{t+n} + \gamma^n Q_{t+n-1}(S_{t+n}, A_{t+n}), \quad n \geq 1, 0 \leq t < T - n.$$