RBE 595 — Reinforcement Learning
Chapter #7 Assignment
n-step Bootstrapping

Arjan Gupta

Problem 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	

Assume discount factor, $\gamma = 0.5$, step size $\alpha = 0.1$ and q_π is initially zero. What are the estimates of $q_\pi(X, U1)$ and $q_\pi(X, U2)$ using 2-step SARSA?

Answer

The estimates of $q_\pi(X, U1)$ and $q_\pi(X, U2)$ using 2-step SARSA are as follows:

Timestep 0

$$\begin{aligned} q_\pi(X,U1) &= q_\pi(X,U1) + \alpha\left[R_{t+1} + \gamma R_{t+2} + \gamma^2 q_\pi(S_{t+2}, A_{t+2}) - q_\pi(X,U1)\right] \\ &= 0 + 0.1\left[16 + 0.5 \cdot 12 + 0.5^2 \cdot 0 - 0\right] \\ &= 0 + 0.1\left[16 + 6 - 0\right] \\ &= 0 + 0.1\left[22\right] \\ &= 0 + 2.2 \\ &= 2.2 \end{aligned}$$

Timestep 1

$$\begin{aligned} q_\pi(X,U2) &= q_\pi(X,U2) + \alpha\left[R_{t+1} + \gamma R_{t+2} + \gamma^2 q_\pi(S_{t+2}, A_{t+2}) - q_\pi(X,U2)\right] \\ &= 0 + 0.1\left[12 + 0.5 \cdot 24 + 0.5^2 \cdot q_\pi(X,U1) - 0\right] \\ &= 0 + 0.1\left[12 + 12 + 0.25 * 2.2\right] \\ &= 0 + 0.1\left[24 + 0.55\right] \\ &= 0 + 0.1\left[24.55\right] \\ &= 2.455 \end{aligned}$$

Timestep 2

$$\begin{aligned} q_\pi(X,U1) &= q_\pi(X,U1) + \alpha\left[R_{t+1} + \gamma R_{t+2} + \gamma^2 q_\pi(S_{t+2}, A_{t+2}) - q_\pi(X,U1)\right] \\ &= 2.2 + 0.1\left[24 + 0.5 \cdot 16 + 0.5^2 \cdot q_\pi(T) - 2.2\right] \\ &= 2.2 + 0.1\left[24 + 8 + 0 - 2.2\right] \\ &= 2.2 + 0.1\left[29.8\right] \\ &= 2.2 + 2.98 \\ &= 5.18 \end{aligned}$$

Problem 1 continued on next page. . .

Timestep 3

$$
\begin{aligned}
q_\pi(X, U1) &= q_\pi(X, U1) + \alpha \left[R_{t+1} + \gamma q_\pi(S_{t+1}, A_{t+1}) - q_\pi(X, U1)\right] \\
&= 5.18 + 0.1 \left[16 + 0.5 \cdot q_\pi(T) - 5.18\right] \\
&= 5.18 + 0.1 \left[16 + 0 - 5.18\right] \\
&= 5.18 + 0.1 \left[10.82\right] \\
&= 6.262
\end{aligned}
$$

Therefore, the estimates of $q_\pi(X, U1)$ and $q_\pi(X, U2)$ using 2-step SARSA are 6.262 and 2.455 respectively.

Problem 2

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

Answer

The purpose of introducing Control Variates in per-decision importance sampling is to further reduce the variance of the estimate of the return.

Plain per-decision importance sampling reduces the variance of the estimate of the return by making sure that the estimate of the whole return is not 0 every time the behavior policy takes an action that the target policy would not have taken. However, the variance of the estimate of the return can be decreased even further by using Control Variates. This is done by using a 'control variate term' in the equation for $G_{t:h}$. Without control variates, the equation for $G_{t:h}$ is as follows:

$$G_{t:h} = R_{t+1} + \gamma G_{t+1:h}, \quad t < h < T$$

With control variates, the equation for $G_{t:h}$ is as follows:

$$G_{t:h} \doteq \rho_t \left[R_{t+1} + \gamma G_{t+1:h}\right] + (1 - \rho_t)V_{h-1}(S_t)\right], \quad t < h < T$$

Where the control variate term is $(1 - \rho_t)V_{h-1}(S_t)$.

This equation uses a 'convex combination' of the estimate of the two terms. This way, when $\rho_t = 0$, we have $G_{t:h} = V_{h-1}(S_t)$, which is the current estimate of the return for the current state. Overall this reduces the 'jaggedness' in the plot of the estimate of the return over time, which reduces the variance of the estimate of the return.

It is also possible to prove that the control variate term does not add bias to the estimate of the return.

Control variates for action-value estimation

With control-estimates, the equation for $G_{t:h}$ in action-value estimation is as follows:

$$G_{t:h} = R_{t+1} + \gamma\rho_{t+1}(G_{t+1:h} - Q(S_{t+1}, A_{t+1})) + \gamma V_{h-1}(S_{t+1})$$

Where $V(S) = \sum_a \pi(a|S)Q(S, a)$.

Here, when $\rho_t = 0$, we have $G_{t:h} = R_{t+1} + \gamma V_{h-1}(S_{t+1})$, instead of simply $G_{t:h} = R_{t+1}$, which would be the case without control variates. This also reduces the variance of the estimate of the return.

Problem 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)?

Answer

The pros and cons of the Tree-Backup algorithm versus off-policy SARSA are as follows:

- **Complexity:** The computational complexity of both the Tree-Backup algorithm and off-policy SARSA over a single episode is $O(n^2)$, where n is the number of steps in the episode. This is because the outer loop of both algorithms iterates over the steps in the episode, and the inner loop of both algorithms are used to calculate iterative sums for the estimate of the return. In the case of off-policy SARSA, one of the inner loops is also used to calculate the importance sampling ratio. For k episodes, the complexity of both algorithms is $O(kn^2)$. Therefore, from a complexity standpoint, neither algorithm is better than the other.
- **Exploration:** Both the Tree-Backup algorithm and off-policy SARSA are off-policy algorithms. Therefore, both algorithms can be used to explore the environment.
- **Variance:** The variance of the Tree-Backup algorithm is lower than that of off-policy SARSA. This is because the off-policy SARSA algorithm uses the importance sampling ratio, which can cause the variance of the estimate of the return to increase, especially when control variates are not used. The Tree-Backup algorithm does not use the importance sampling ratio, and therefore does not have this problem.
- **Bias:** As a trade-off for lower variance, the Tree-Backup algorithm has higher bias than off-policy SARSA.
- **Others:** One benefit of using Tree-Backup algorithm over off-policy SARSA is that the Tree-Backup algorithm can be used when we have no knowledge of the underlying distribution of the behavior policy. This can be useful depending on the application, for example, if the behavior policy is a human, and we have no knowledge of the human's decision-making process.

Problem 4

(Exercise 7.4) Prove that the n-step return of Sarsa (7.4) can be written exactly in terms of a novel TD error, as

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

Answer

The n-step return of Sarsa (7.4) is as follows:

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

We can rewrite this as follows:

$$\begin{aligned} G_{t:t+n} &= R_{t+1} + \gamma R_{t+2} + \ldots + \gamma^{n-1} R_{t+n} + \gamma^n Q_{t+n-1}(S_{t+n}, A_{t+n}) \\ &= R_{t+1} + \gamma R_{t+2} + \ldots + \gamma^{n-1} R_{t+n} + \gamma^n Q_{t+n-1}(S_{t+n}, A_{t+n}) - \gamma^n Q_{t+n-1}(S_{t+n}, A_{t+n}) + \gamma^n Q_{t+n-1}(S_{t+n}, A_{t+n}) \end{aligned}$$