RBE595 - Week 9 Assignment

Keith Chester

Due date: March 13, 2023

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?

Let SAR = []
t=0
Do $A_0 = U1 \rightarrow R_1 = 16$ $S_{0+1} = X$
SAR $\leftarrow$ ((X, $U1$), 16)
$S_{0+1} \neq$ terminal; $A_1 = \pi(\cdot|S_1) = U2$
t=1
Do $A_1 = U2 \rightarrow R_2 = 12$ $S_{1+1} = X$
SAR[((X, $U1$), 16)] $\leftarrow$ ((X, $U2$), 12)
$S_{1+1} \neq$ terminal; $A_2 = \pi(\cdot|S_{1+1}) = U1$
$t = 1 - 2 + 1 = 0$
$G = \sum_{i=0+1}^{min(2,\infty)} \gamma^{i-0-1} R_i = 6$
$t + n < T$; $G = G + \gamma^n Q(S_{t+n}, A_{t+n}) = 0.6$
t=2
Do $A_2 = U1 \rightarrow R_3 = 24$ $S_{2+1} = X$
SAR[((X, $U1$), 16),((X, $U2$), 12)] $\leftarrow$ ((X, $U3$), 24)
$S_{2+1} \neq$ terminal; $A_3 = \pi(\cdot|S_{2+1}) = U1$
$t = 2 - 2 + 1 = 1$
$G = \sum_{i=1+1}^{min(3,\infty)} \gamma^{i-0-1} R_i = 12$
$t + n < T$; $G = G + \gamma^n Q(S_{t+n}, A_{t+n}) = 1.215$
t=3
Do $A_3 = U1 \rightarrow R_4 = 16$ $S_{3+1} = T$
$S_{3+1} =$ terminal; $T = t + 1 = 4$;
$t = 3 - 2 + 1 = 2$
$G = \sum_{i=0+1}^{min(2,\infty)} \gamma^{i-0-1} R_i = 12$
$t + n = T$, no adjustment in G
$Q(S_t, A_t) = Q(S_2, A_2) = Q(X, U1) = .1 * (8 + .6) = 0.86$

Problem 2

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

Importance sampling in reinforcement learning can lead to high variance in estimations, and thus create wild swings in performance of a model. When we use per-decision important sampling we apply control variates as a method of reducing the variance of an estimator. The control variate is a variable that correlates with a variable of interest but has a known expectation; we can then adjust our estimates to reduce variance.

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)?

Let's look at each algorithm one at a time:

Tree-Backup Algorithm

Pros:

- Tree-Backup Algorithm can handle delayed/sparse rewards allowing it to work in environments with more complex sequences before rewards.
- Its exploration is efficient; when it updates it selects values of state/action pairs that are relevant, so less overall work needed to learn
- It can return estimates of the value function with a low bias

Cons:

- TBA can have high variance since it involves a high number of random samples
- The algorithm is difficult to implement and requires significant computation compared to comparable approaches.

Off-policy SARSA

:

Pros:

- Off-policy SARSA has lower variance than TBA, especially when the target policy is closer to the behaviour policy. This is due to importance sampling when updating the value function.
- Off-policy SARSA is simpler and more computationally efficient than TBA

Cons:

- Off-policy SARSA only updates the value function based on the very next action taken by the target policy, instead of all possible future actions. This results in two major downsides: it is worse at solving environments which delays rewards until a large amount of actions are taken, and also introduces a higher bias in value function estimation.
- Compared to TBA, this algorithm is less efficient in exploration as it doesn't selectively update the values of certain state/action pairs based on the current state/action pair.

In summary, when you are faced with a problem that requires dealing with delayed rewards, reach for the more complex Tree-Backup Algorithm over Off-policy SARSA.

Problem 4

Exercise 7.4 of the textbook, page 148.

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

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

For this, we must look at the min function in our summation - this leads us to two possibilities. First we'll look at $t + n < T$, then later we'll look at the other case. If we hold $t + n < T$ to be true, then the minimum function becomes:

$$G_{t:t+n} = Q_{t-1}(S_t, A_t) + \sum_{k=t}^{t+n-1} \gamma^{k-t} \Big[R_{k+1} + \gamma Q_k(S_{k+1}, A_{k+1}) - Q_{k-1}(S_k, A_k) \Big] \tag{2}$$

...which we can begin to expand upon:

$$\begin{aligned} G_{t:t+n} = Q_{t-1}(S_t, A_t) + \\ \gamma^0 \Big[R_{t+1} + \gamma Q_t(S_{t+1}, A_{t+1}) - Q_{t-1}(S_t, A_t) \Big] + \\ \gamma^1 \Big[R_{t+2} + \gamma Q_{t+1}(S_{t+2}, A_{t+2}) - Q_t(S_t + 1, A_t + 1) \Big] + \\ \vdots \\ \gamma^{n-1} \Big[R_{t+n} + \gamma Q_k(S_{t+n}, A_{t+n}) - Q_{t+n-2}(S_{t+n-1}, A_{t+n-1}) \Big] \end{aligned} \tag{3}$$

We can simplify this expanded sequence by eliminating first our Q terms, which completely cancel out save $Q_k(S_{t+n}, A_{t+n})$; this leaves us with a SARSA n-step return:

$$G_{t:t+n} = R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{n-1} R_{t+n} + \gamma^n Q_{t+n-1}(S_{t+n}, A_{t+n}) \tag{4}$$

Now let's look at the other case, when $t + n > T$:

$$G_{t:t+n} = Q_{t-1}(S_t, A_t) + \sum_{k=t}^{T-1} \gamma^{k-t} \Big[R_{k+1} + \gamma Q_k(S_{k+1}, A_{k+1}) - Q_{k-1}(S_k, A_k) \Big] \tag{5}$$

$$\begin{aligned} G_{t:t+n} = Q_{t-1}(S_t, A_t) + \\ \gamma^0 \Big[R_{t+1} + \gamma Q_t(S_{t+1}, A_{t+1}) - Q_{t-1}(S_t, A_t) \Big] + \\ \gamma^1 \Big[R_{t+2} + \gamma Q_{t+1}(S_{t+2}, A_{t+2}) - Q_t(S_t + 1, A_t + 1) \Big] + \\ \vdots \\ \gamma^{n-1} \Big[R_{t+n} + \gamma Q_{T-1}(S_T, A_T) - Q_{T-2}(S_{T-1}, A_{T-1}) \Big] \end{aligned} \tag{6}$$

Once again we see that Q terms cancel out save for $Q_{T-1}(S_T, A_T)$, leaving us with the SARSA n-step return function as we set out to prove.