Homework 4

ELEN E6885: Introduction to Reinforcement Learning

Due: December 7, 2022

Problem 1 (***n*-Step Return, 15 Points**)

The expected value of all n-step returns is guaranteed to improve in a certain way over the current value function as an approximation to the true value function. Prove the following *error reduction property* of n-step returns

$$\max_s \left| E_\pi \left[G_t^{(n)} \middle| S_t = s \right] - V_\pi(s) \right| \leq \gamma^n \max_s \left| V_t(s) - V_\pi(s) \right|,$$

where $G_t^{(n)}$ is n-step return at time t.

Proof. For any start state s, the following holds:

$$\begin{aligned}
&\left| E_\pi \left[G_t^{(n)} | S_t = s \right] - V_\pi(s) \right| \\
=&\Bigg| E_\pi \left[R_{t+1} + \cdots + \gamma^{n-1} R_{t+n} + \gamma^n V_t(S_{t+n}) \middle| S_t = s \right] \\
&-E_\pi \left[R_{t+1} + \cdots + \gamma^{n-1} R_{t+n} + \gamma^n \sum_{k=0}^{\infty} \gamma^k R_{t+n+k+1} \middle| S_t = s \right] \Bigg| \\
=&\gamma^n \left| \sum_{s'} P_\pi(S_{t+n} = s' | S_t = s) \left(V_t(s') - V_\pi(s') \right) \right| \\
\leq&\gamma^n \max_s \left| V_t(s) - V^\pi(s) \right|.
\end{aligned}$$

Therefore, by maximizing over all states on the left hand side of the above inequality, we have

$$\max_s \left| E_\pi \left[G_t^{(n)} \middle| S_t = s \right] - V_\pi(s) \right| \leq \gamma^n \max_s \left| V_t(s) - V_\pi(s) \right|.$$

Problem 2 (**On-line vs. Off-line Update, 35 Points**)

To distinguish two different ways of making updates in reinforcement learning algorithms (i.e., on-line and off-line updating), answer the following questions.

1. [5 pts] What is the difference between on-line and off-line updating methods?

2. [5 pts] In all following questions, consider an episode: $A, +1, B, +2, A, +1, T$ from an *undiscounted* MDP, where A, B are two non-terminal states, T is the terminal state and the number after each state is an immediate reward. Using a learning rate of $\alpha = 0.1$, and assuming initial state values of 0. What is the total update to $V(A)$ *on-line* every-visit constant-α Monte Carlo method makes after the episode finishes? What about *off-line* every-visit constant-α Monte Carlo method?

3. [5 pts] What is the total update to $V(A)$ *on-line* TD(0) method makes after the episode finishes? What about *off-line* TD(0) method?

4. [10 pts] Assume $\lambda = 0.5$. What is the total update to $V(A)$ *on-line* forward-view TD(λ) method makes after the episode finishes? What about *off-line* forward-view TD(λ) method?

5. [10 pts] Assume $\lambda = 0.5$. What is the total update to $V(A)$ *on-line* backward-view TD(λ) method makes after the episode finishes? What about *off-line* backward-view TD(λ) method?

Solution.

1. In on-line updating, the updates are done during the episode, as soon as the increment is computed. In off-line updating, on the other hand, the increments are accumulated "on the side" and are not used to change value estimates until the end of the episode.

2. The updates to $V(A)$ within the episode using on-line constant-α Monte Carlo method are

$$V(A) = V(A) + \alpha\,(4 - V(A)) = 0 + 0.1 \times 4 = 0.4,$$
$$V(A) = V(A) + \alpha\,(1 - V(A)) = 0.4 + 0.1 \times 0.6 = 0.46.$$

So the total update to $V(A)$ using on-line constant-α Monte Carlo method is 0.46.

The increments of $V(A)$ within the episode using off-line constant-α Monte Carlo method are

$$\Delta V(A) = \alpha\,(4 - V(A)) = 0.1 \times 4 = 0.4,$$
$$\Delta V(A) = \alpha\,(1 - V(A)) = 0.1 \times 1 = 0.1.$$

So the total update to $V(A)$ using off-line constant-α Monte Carlo method is 0.5..

3. The updates to $V(A)$ within the episode using on-line TD(0) method are

$$V(A) = V(A) + \alpha\,(1 + V(B) - V(A)) = 0 + 0.1 \times 1 = 0.1,$$
$$V(A) = V(A) + \alpha\,(1 + V(T) - V(A)) = 0.1 + 0.1 \times 0.9 = 0.19.$$

So the total update to $V(A)$ using on-line TD(0) method is 0.19.

The increments of $V(A)$ within the episode using off-line TD(0) method are

$$\Delta V(A) = \alpha\,(1 + V(B) - V(A)) = 0.1 \times 1 = 0.1,$$
$$\Delta V(A) = \alpha\,(1 + V(T) - V(A)) = 0.1 \times 1 = 0.1.$$

So the total update to $V(A)$ using off-line TD(0) method is 0.2.

4. We first calculate the λ-return from state A at $t = 0$ and $t = 2$, respectively, as follows:

$$\begin{aligned} G_0^\lambda &= (1-\lambda)G_0^{(1)} + (1-\lambda)\lambda G_0^{(2)} + \lambda^2 G_0^{(3)} \\ &= 0.5 \times 1 + 0.25 \times 3 + 0.25 \times 4 = 2.25, \\ G_2^\lambda &= G_2^{(1)} = 1. \end{aligned}$$

The updates to $V(A)$ within the episode using on-line forward-view TD(λ) are

$$V(A) = V(A) + \alpha\left(G_0^\lambda - V(A)\right) = 0 + 0.1 \times 2.25 = 0.225,$$
$$V(A) = V(A) + \alpha\left(G_2^\lambda - V(A)\right) = 0.225 + 0.1 \times 0.775 = 0.3025.$$

So the total update to $V(A)$ using on-line forward-view TD(λ) is 0.3025.

The increments of $V(A)$ within the episode using off-line forward-view TD(λ) are

$$\Delta V(A) = \alpha\left(G_0^\lambda - V(A)\right) = 0.1 \times 2.25 = 0.225,$$
$$\Delta V(A) = \alpha\left(G_2^\lambda - V(A)\right) = 0.1 \times 1 = 0.1.$$

So the total update to $V(A)$ using off-line forward-view TD(λ) is 0.325.

5. The eligibility trace for state A at $t = 0$ and $t = 2$ is $E_0(A) = 1, E_2(A) = 1.25$, respectively.

The updates to $V(A)$ within the episode using on-line backward-view TD(λ) are

$$V(A) = V(A) + \alpha\,(1 + V(B) - V(A))\,E_0(A) = 0 + 0.1 \times 1 \times 1 = 0.1,$$
$$V(A) = V(A) + \alpha\,(1 + V(T) - V(A))\,E_2(A) = 0.1 + 0.1 \times 0.9 \times 1.25 = 0.2125.$$

So the total update to $V(A)$ using on-line backward-view TD(λ) is 0.2125.

The updates to $V(A)$ within the episode using off-line backward-view TD(λ) are

$$\Delta V(A) = \alpha\,(1 + V(B) - V(A))\,E_0(A) = 0 + 0.1 \times 1 \times 1 = 0.1,$$
$$\Delta V(A) = \alpha\,(1 + V(T) - V(A))\,E_2(A) = 0.1 + 0.1 \times 1 \times 1.25 = 0.225.$$

So the total update to $V(A)$ using off-line backward-view TD(λ) is 0.325.

Problem 3 (Forward vs. Backward view of TD(λ), 25 Points)

We know that when using off-line updates, forward-view and backward-view TD(λ) are equivalent, i.e., the total update to a value function at the end of an episode is the same. In other words, the off-line TD(λ) (i.e., backward view) exactly matches the off-line λ-return algorithm (i.e., forward view).

1. [20 pts] As a special case, follow the steps below to prove that off-line TD(1) (i.e., backward view) and off-line every-visit constant-α Monte Carlo method (i.e., forward view) are equivalent.

   a. [5 pts] Consider an episode which terminates after T steps. Prove that the original statement is equivalent to show that for any state s,

   $$\sum_{t=0}^{T-1} \alpha\delta_t E_t(s) = \sum_{t=0}^{T-1} \alpha\left(G_t - V(S_t)\right) \mathbf{1}\left(S_t = s\right), \tag{1}$$

   where $\mathbf{1}(\cdot)$ is the indicator function, which equals to 1 if $S_t = s$ and 0 otherwise.

   b. [5 pts] For any $0 \leq t \leq T-1$ and state s, prove that the accumulating eligibility trace can be written explicitly as

   $$E_t(s) = \sum_{k=0}^{t} \gamma^{t-k} \cdot \mathbf{1}\left(S_k = s\right). \tag{2}$$

   c. [10 pts] Prove that the equality in (1) holds by plugging (2) into the left-hand-side of (1).

2. [5 pts] Is it possible to construct a version of on-line TD(λ) method (i.e., backward view) that matches the on-line λ-return algorithm (i.e., forward view) exactly? Explain your answer.

Solution.

1a. The goal is to show that over the course of an episode that terminates after T steps, the total update to the value of state s for TD(1) is the same as total update for every-visit constant-α Monte Carlo method. The total update to state s for TD(1) is

$$\sum_{t=0}^{T-1} \alpha\delta_t E_t(s),$$

and the total update for every-visit constant-α Monte Carlo is

$$\sum_{t=0}^{T-1} \alpha\left(G_t - V(S_t)\right) \mathbf{1}\left(S_t = s\right).$$

Therefore, the original statement is equivalent to (1).

1b. For any $0 \leq t \leq T-1$, let $0 \leq t_1, t_2, \cdots, t_n \leq t$ denote the time when state s has been visited. According to the definition of the accumulating trace, the i-th visit of state s at step t_i will increase the trace by 1. And since $\lambda = 1$, this increment will decay by γ^{t-t_i} when reaching step t. Therefore, we have

$$E_t(s) = \sum_{i=1}^{n} 1 \cdot \gamma^{t-t_i} = \sum_{k=0}^{t} \gamma^{t-k} \cdot \mathbf{1}(S_k = s).$$

1c. By plugging (2) into the left-hand-side of (1), we have

$$\begin{aligned}\sum_{t=0}^{T-1} \alpha\delta_t E_t(s) &= \sum_{t=0}^{T-1} \alpha\delta_t \sum_{k=0}^{t} \gamma^{t-k} \cdot \mathbf{1}(S_k = s)\\ &\overset{(a)}{=} \sum_{k=0}^{T-1} \mathbf{1}(S_k = s) \sum_{t=k}^{T-1} \alpha\delta_t\gamma^{t-k}\\ &\overset{(b)}{=} \sum_{k=0}^{T-1} \mathbf{1}(S_k = s) \sum_{t=k}^{T-1} \alpha\gamma^{t-k}\left(R_{t+1} + \gamma V(S_{t+1}) - V(S_t)\right), \end{aligned} \tag{3}$$

where (a) is by interchanging the order of summation, and (b) is by plugging in the definition of one-step TD error. Since

$$\begin{aligned}&\sum_{t=k}^{T-1} \alpha\gamma^{t-k}\left(R_{t+1} + \gamma V(S_{t+1}) - V(S_t)\right)\\ &= \sum_{t=k}^{T-1} \alpha\gamma^{t-k} R_{t+1} + \sum_{t=k}^{T-1} \alpha\gamma^{t-k}\gamma V(S_{t+1}) - \sum_{t=k}^{T-1} \alpha\gamma^{t-k} V(S_t)\\ &= \alpha G_k + \sum_{l=k+1}^{T} \alpha\gamma^{l-k} V(S_l) - \sum_{t=k}^{T-1} \alpha\gamma^{t-k} V(S_t)\\ &= \alpha(G_k - V(S_k)),\end{aligned}$$

from (3) we have

$$\begin{aligned}\sum_{t=0}^{T-1} \alpha\delta_t E_t(s) &= \sum_{k=0}^{T-1} \alpha(G_k - V(S_k)) \cdot \mathbf{1}(S_k = s)\\ &= \sum_{t=0}^{T-1} \alpha(G_t - V(S_t)) \cdot \mathbf{1}(S_t = s).\end{aligned}$$

2. No version of on-line TD(λ) can be constructed that matches the on-line λ-return algorithm exactly. Because the λ-return uses rewards and states beyond the current time step.

Problem 4 (Linear Function Approximation, 25 Points)

Consider the small corridor gridworld shown in Fig. 1 below. S and G represents the start and goal (terminal) state, respectively. In each of the two non-terminal states, there are only two actions, *right* and *left*. These actions have their usual consequences in the start state (left causes no movement in the start state). But in the middle state they are reversed, so that *right* moves to the left and *left* moves to the right. The reward is -1 per step as usual. We approximate the action-value function using two features $\mathrm{x}_1(s,a) = \mathbf{1}\,(a = \mathit{right})$ and $\mathrm{x}_2(s,a) = \mathbf{1}\,(a = \mathit{left})$ for all state-action pair (s,a). We sample an episode till the goal by sequentially taking actions *right*, *right*, *right*, *left*. Assume the experiment is *undiscounted*.

1. [5 pts] Approximate the action-value function by a linear combination of these features with two parameters: $\hat{q}(s,a,\mathbf{w}) = \mathrm{x}_1(s,a)w_1 + \mathrm{x}_2(s,a)w_2$. If $w_1 = w_2 = 1$, calculate the λ-return q_t^λ corresponding to this episode for $\lambda = 0.5$.

2. [5 pts] Using the forward-view TD(λ) algorithm with off-line updates and our linear function approximator, what are the sequence of updates to weight w_1? What is the total update to weight w_1? Use $\lambda = 0.5, \gamma = 1, \alpha = 0.5$ and start with $w_1 = w_2 = 1$.

3. [5 pts] Define the TD(λ) accumulating eligibility trace $\mathbf{e}_t$ when using linear value function approximation. Write down the sequence of eligibility traces corresponding to *right* action, using $\lambda = 0.5, \gamma = 1$.

4. [5 pts] Using the backward-view TD(λ) algorithm with off-line updates and our linear function approximator, what are the sequence of updates to weight w_1? What is the total update to weight w_1? Use $\lambda = 0.5, \gamma = 1, \alpha = 0.5$ and start with $w_1 = w_2 = 1$.

5. [5 pts] Based on your results in previous questions, when using off-line updates and linear function approximation, are forward-view and backward-view TD(λ) equivalent to each other?

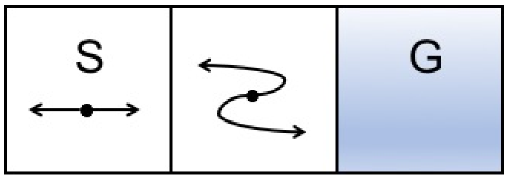

Figure 1: Small corridor gridworld

Solution.

1. The sequence of approximate action-values corresponding to this episode is all 1. Then the λ-returns for this episode are as follows:

$$
\begin{aligned}
q_1^\lambda &= 0.5\left((-1+1) + 0.5\times(-2+1) + 0.5^2\times(-3+1)\right) + 0.5^3\times(-4) = -1\\
q_2^\lambda &= 0.5\left((-1+1) + 0.5\times(-2+1)\right) + 0.5^2\times(-3) = -1\\
q_3^\lambda &= 0.5\,(-1+1) + 0.5\times(-2) = -1\\
q_4^\lambda &= -1.
\end{aligned}
$$

2. The updates to weight w_1 in forward TD(λ) are as follows:

$$\begin{aligned}
\Delta w_1^1 &= \alpha \left(q_1^\lambda - \hat{Q}(s,a,\mathbf{w})\right) \mathbf{x}_1(s,a) = 0.5(-1-1)1 = -1 \\
\Delta w_1^2 &= \alpha \left(q_2^\lambda - \hat{Q}(s,a,\mathbf{w})\right) \mathbf{x}_1(s,a) = 0.5(-1-1)1 = -1 \\
\Delta w_1^3 &= \alpha \left(q_3^\lambda - \hat{Q}(s,a,\mathbf{w})\right) \mathbf{x}_1(s,a) = 0.5(-1-1)1 = -1 \\
\Delta w_1^4 &= \alpha \left(q_4^\lambda - \hat{Q}(s,a,\mathbf{w})\right) \mathbf{x}_1(s,a) = 0.5(-1-1)0 = 0.
\end{aligned}$$

And the total update for w_1 is -3.

3. The accumulating eligibility trace is $\mathbf{e}_t = \gamma\lambda\mathbf{e}_{t-1} + \mathbf{x}(s,a)$. The sequence of eligibility traces corresponding to *right* action is $1, \frac{3}{2}, \frac{7}{4}$, and $\frac{7}{8}$.

4. The updates to weight w_1 in backward TD(λ) are as follows:

$$\begin{aligned}
\Delta w_1^1 &= \alpha\delta_1 e_1 = 0.5(-1+1-1)1 = -\frac{1}{2} \\
\Delta w_1^2 &= \alpha\delta_2 e_2 = 0.5(-1+1-1)\frac{3}{2} = -\frac{3}{4} \\
\Delta w_1^3 &= \alpha\delta_3 e_3 = 0.5(-1+1-1)\frac{7}{4} = -\frac{7}{8} \\
\Delta w_1^4 &= \alpha\delta_4 e_4 = 0.5(-1+0-1)\frac{7}{8} = -\frac{7}{8}.
\end{aligned}$$

And the total update for w_1 is -3.

5. Yes, forward-view and backward-view TD(λ) with off-line updates and linear function approximation are equivalent.