CMPUT 609 Assignment 4

Shaurya Seth

February 19, 2023

Question 1

The TD error is given by:

$$\delta_t = R_{t+1} + \gamma V(S_{t+1}) - V(S_t)$$

The n-step error can be written as a sum of TD errors:

$$\begin{aligned} G_{t:t+n} - V(S_t) &= R_{t+1} + \gamma G_{t+1:t+n} - V(S_t) - \gamma V(S_{t+1}) + \gamma V(S_{t+1}) \\ &= \delta_t + \gamma(G_{t+1:t+n} - V(S_{t+1})) \\ &= \delta_t + \gamma\delta_{t+1} + \gamma^2\delta_{t+2} \cdots + \gamma^{n-1}\delta_{t+n-1} \\ &= \sum_{k=t}^{t+n-1} \gamma^{k-t}\delta_k \end{aligned}$$

Question 2

A larger random walk task was used to make the difference between the performances of one-step TD and Monte Carlo much larger. This better highlights the advantage of using n-step methods. A smaller walk would have shifted the advantage to a smaller value of n. Similarly, a larger walk would shift the advantage to a larger n.

Before the initial value estimates were 0.5 and the left-side outcome was 0. Making the initial estimates 0 and the left-side outcome -1 gives a wider range of learning for the methods and again better highlights the advantage of using n-step methods. This probably did not change the best value of n but made the differences between different n values more apparent.

Question 3

We use the fact that the importance sampling ratio has an expected value of one:

$$\begin{aligned} \mathbb{E}[G_{t:h}] &= \mathbb{E}[R_{t+1} + \gamma\rho_{t+1}(G_{t+1:h} - Q_{h-1}(S_{t+1}, A_{t+1}) + \gamma\bar{V}_{h-1}(S_{t+1})] \\ &= \mathbb{E}[R_{t+1}] + \mathbb{E}[\gamma\rho_{t+1}(G_{t+1:h} - Q_{h-1}(S_{t+1}, A_{t+1})] + \mathbb{E}[\gamma\bar{V}_{h-1}(S_{t+1})] \\ &= R_{t+1} + \mathbb{E}[\gamma G_{t+1:h} - \gamma Q_{h-1}(S_{t+1}, A_{t+1}] + \gamma\bar{V}_{h-1}(S_{t+1}) \\ &= R_{t+1} + \mathbb{E}[G_{t:h} - R_{t+1} - \gamma Q_{h-1}(S_{t+1}, A_{t+1}] + \gamma\bar{V}_{h-1}(S_{t+1}) \\ &= \mathbb{E}[G_{t:h}] - \gamma\mathbb{E}[Q_{h-1}(S_{t+1}, A_{t+1})] + \gamma\bar{V}_{h-1}(S_{t+1}) \\ &= \mathbb{E}[G_{t:h}] \end{aligned}$$

Question 4

The off-policy version of the n-step return can be written as:

$$\begin{aligned} G_{t:h} - V(S_t) &= \rho_t(R_{t+1} + \gamma G_{t+1:t+n}) + (1 - \rho_t)V_{h-1}(S_t) \\ &= \rho_t(R_{t+1} + \gamma G_{t+1:t+n} - V(S_t)) \\ &= \rho_t(\delta_t + \gamma(G_{t+1:t+n} - V(S_{t+1}))) \\ &= \rho_t \sum_{k=t}^{h} \gamma^{k-t}\delta_k \end{aligned}$$

Question 5

Not attempted.

Question 6

The equation of n-step off-policy TD can be converted to semi-gradient form as:

$$\mathbf{w}_{t+n} = \mathbf{w}_{t+n-1} + \alpha\rho_{t:t+n-1}[G_{t:t+n} - \hat{v}(S_t, \mathbf{w}_{t+n-1})]\nabla\hat{v}(S_t, \mathbf{w}_{t+n-1}), \qquad 0 \leq t < T$$

Here the n-step return is given by:

$$G_{t:t+n} = R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{n-1}R_{t+n} + \gamma^n\hat{v}(S_t, \mathbf{w}_{t+n}), \qquad n \geq 1 \text{ and } 0 \leq t < T - n$$

with $G_{t:t+n} = G_t$ if $t + n \geq T$.

Question 7

The equation of n-step Q(σ) can be converted to semi-gradient form as:

$$\mathbf{w}_{t+n} = \mathbf{w}_{t+n-1} + \alpha[G_{t:t+n} - \hat{q}(S_t, A_t, \mathbf{w}_{t+n-1})]\nabla\hat{q}(S_t, A_t, \mathbf{w}_{t+n-1}), \qquad 0 \leq t < T$$

The n-step return is given by:

$$\begin{aligned} G_{t:h} = R_{t+1} &+ \gamma(\sigma_{t+1}\rho_{t+1} + (1 - \sigma_{t+1})\pi(A_{t+1}|S_{t+1}))(G_{t+1:h} - \hat{q}(S_{t+1}, A_{t+1}, \mathbf{w}_{h-1})) \\ &+ \gamma\sum_a \pi(a|S_{t+1})\hat{q}(S_t, a, \mathbf{w}_{t+n}) \end{aligned}$$

with $t < h \leq T$ and $G_{h:h} = \hat{q}(S_h, A_h, \mathbf{w}_{h-1})$ if $h < T$ or $G_{T-1:T} = R_T$ if $h = T$.

Question 8

Monte Carlo methods are an extreme case of n-step methods where $n = T$. The pseudocode would just be a simpler version of semi-gradient n-step Sarsa. Also Monte Carlo would have high variance and computational costs so it makes sense to exclude it. They would have similar performance on the Mountain Car task as the large n cases in Figure 10.4.

Question 9

The results in Figure 10.4 have a higher standard error for larger n because the algorithm has to wait longer before making an update whereas for a smaller n updates are made more frequently. In other words, variance grows as n becomes larger.

Question 10

Pseudocode for a differential version of semi-gradient Q-learning:

Input: a differentiable action-value function parameterization $\hat{q} : \mathcal{S} \times \mathcal{A} \times \mathbb{R}^d \to \mathbb{R}$
Algorithm parameters: step sizes $\alpha, \beta > 0$, small $\varepsilon > 0$
Initialize value-function weights $\mathbf{w} \in \mathbb{R}^d$ arbitrarily (e.g., $\mathbf{w} = 0$)
Initialize average reward estimate $\bar{R} \in \mathbb{R}$ arbitrarily (e.g., $\bar{R} = 0$)

Initialize state S, and action A
Loop for each step:
    Choose A as a function of $\hat{q}(S, \cdot, \mathbf{w})$ (e.g., ε-greedy)
    Take action A, observe R, S'

$\delta \leftarrow R - \bar{R} + \text{argmax}_a \hat{q}(S', A', \mathbf{w}) - \hat{q}(S, A, \mathbf{w})$
$\bar{R} \leftarrow \bar{R} + \beta\delta$
$\mathbf{w} \leftarrow \mathbf{w} + \alpha\delta\nabla\hat{q}(S, A, \mathbf{w})$
$S \leftarrow S'$
$A \leftarrow A'$

Question 11

For the differential version of TD(0) we need to include the following equations:

$$\bar{R}_{t+1} = \bar{R}_t + \beta\delta_t$$
$$\mathbf{w}_{t+1} = \mathbf{w}_t + \alpha\delta_t\nabla\hat{v}(S_t, \mathbf{w}_t)$$

Question 12

The average reward is given by:

$$r(\pi) = \lim_{h\to\infty} \frac{1}{h} \sum_{t=1}^{h} R_t = 0$$

The values of the states are given by:

$$v_\pi(a) = \lim_{\gamma\to 1} \lim_{h\to\infty} \sum_{t=0}^{h} \gamma^t(-1)^t = \lim_{\gamma\to 1} \frac{1}{1+\gamma} = 0.5$$
$$v_\pi(b) = \lim_{\gamma\to 1}(-1 + \gamma v_\pi(a)) = -0.5$$

Question 13

Using (3) we get:

$$\mu_\pi(s) = 1$$

Using (2) we get:

$$v_\pi(a) + v_\pi(b) = 0$$

Using the Bellman equations we get:

$$v_\pi(b) = -1 + \gamma v_\pi(a)$$

Solving them together and taking the limit gives us $v_\pi(a) = 0.5$ and $v_\pi(b) = -0.5$.

Question 14

Since the MDP is deterministic:

$$\mu_\pi(s) = 1$$

The average reward is:

$$r(\pi) = \frac{4}{3}$$

Using (2) we get:

$$v_\pi(a) + v_\pi(b) + v_\pi(c) = 0$$

Using the Bellman equations we get:

$$v_\pi(a) = 1 - r(\pi) + v_\pi(b)$$
$$v_\pi(b) = 1 - r(\pi) + v_\pi(c)$$
$$v_\pi(c) = 2 - r(\pi) + v_\pi(a)$$

Solving them together we get $v_\pi(a) = -\frac{1}{3}$, $v_\pi(b) = 0$ and $v_\pi(c) = \frac{1}{3}$.

Question 15

The sequence of $R_{t+1} - \bar{R}_t$ errors would be:

$$-\frac{1}{3}, -\frac{1}{3}, \frac{2}{3}, -\frac{1}{3}, \dots$$

The sequence of δ_t errors would be:

$$0, 0, 0, 0, \dots$$

The second error sequence would produce a more stable estimate of the average reward if the estimate were allowed to change in response to the errors. This is because the δ_t errors are zero when the state-value estimates are correct.

Question 16

We can change β with β_n:

$$\beta_n = \beta / \bar{o}_n$$

$$\bar{o}_n = \bar{o}_{n-1} + \beta(1 - \bar{o}_{n-1})$$

for $n \geq 0$, with $\bar{o}_0 = 0$.