1 On-policy Control with Approximation

1.1 Exercise 10.1

1.1.1 Q

We have not explicitly considered or given pseudocode for any Monte Carlo methods or in this chapter. What would they be like? Why is it reasonable not to give pseudocode for them? How would they perform on the Mountain Car task?

1.1.2 A

- Monte Carlo is n-step Sarsa with $n \to \infty$
- This is same pseudocode as given, but with full episodes and G_t rather than $G_{t:t+n}$.
- Could have been very poor on the mountain car as may never have finished the first episode and does not learn within an episode (online)

1.2 Exercise 10.2

1.2.1 Q

Give pseudocode for semi-gradient one-step *Expected Sarsa* for control.

1.2.2 A

Expected sarsa is the same but the target is

$$\sum_{k=t}^{t+n-1} \gamma^{k-t} R_{k+1} + \sum_a \pi(a|S_{t+n}) q_{t+n-1}(S_{t+n}, a)$$

1.3 Exercise 10.3

1.3.1 Q

Why do the results shown in Figure 10.4 have higher standard errors at large n than at small n?

1.3.2 A

The longer the step length then the greater the variance in initial runs, this is because the agent needs to wait for n steps to start learning. Some initial episodes of high n cases could have been very poor.

1.4 Exercise 10.4

1.4.1 Q

Give pseudocode for a differential version of semi-gradient Q-learning.

1.4.2 A

Same as others but with the target

$$R_{t+1} - \bar{R}_{t+1} - \max_a \hat{q}(S_{t+1}, a, \mathbf{w}_t)$$

1.5 Exercise 10.5

1.5.1 Q

What equations are needed (beyond 10.10) to specify the differential version of TD(0)?

1.5.2 A

Just need the semi-gradient update

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

where

$$\delta_t = R_{t+1} - \bar{R}_{t+1} + \hat{v}(S_{t+1}, \mathbf{w}_t) - \hat{v}(S_t, \mathbf{w}_t)$$

1.6 Exercise 10.6

1.6.1 Q

Consider a Markov reward process consisting of a ring of three states A, B, and C, with state transitions going deterministically around the ring. A reward of 1 is received upon arrival in A and otherwise the reward is 0. What are the differential values of the three states?

1.6.2 A

The average reward is $\bar{R} = \frac{1}{3}$. To calculate the differential return we have

$$V(A) = \sum_t (a_t - \bar{R})$$

where $a_i = \mathbb{1}\{i + 1 \equiv 0 \pmod 3\}$. This doesn't converge in the normal way, so to attempt to calculate it let's consider

$$V(A; \gamma) = \sum_t \gamma^t \left(a_t - \frac{1}{3}\right)$$

then, formally, we have

$$\lim_{\gamma \to 1} V(A; \gamma) = V(A).$$

Now

$$\begin{aligned} V(A; \gamma) &= -\frac{1}{3} - \frac{1}{3}\gamma + \frac{2}{3}\gamma^2 + \sum_{t=3}^{\infty} \gamma^t \left(a_t - \frac{1}{3}\right) \\ &= \frac{1}{3}(2\gamma^2 - \gamma - 1) + \gamma^3 \sum_{t=0}^{\infty} \gamma^t \left(a_t - \frac{1}{3}\right) \end{aligned}$$

so

$$\begin{aligned} V(A; \gamma) &= \frac{1}{3}\frac{2\gamma^2 - \gamma - 1}{1 - \gamma^3} \\ &= -\frac{1}{3}\frac{2\gamma + 1}{\gamma^2 + \gamma + 1} \end{aligned}$$

which leads to $V(A) = -\frac{1}{3}$.

Then we have

$$V(A) = -\frac{1}{3} + V(B) \quad \implies \quad V(B) = 0$$

and

$$V(B) = -\frac{1}{3} + V(C) \quad \implies \quad V(C) = \frac{1}{3}.$$

1.7 Exercise 10.7

1.7.1 Q

Suppose there is an MDP that under any policy produces the deterministic sequence of rewards 1, 0, 1, 0, 1, 0, . . . going on forever. Technically, this is not allowed because it violates ergodicity; there is no stationary limiting distribution μ_π and the limit (10.7) does not exist. Nevertheless, the average reward (10.6) is well defined; What is it? Now consider two states in this MDP. From A, the reward sequence is exactly as described above, starting with a 1, whereas, from B, the reward sequence starts with a 0 and then continues with 1, 0, 1, 0, ... The differential return (10.9) is not well defined for this case as the limit does not exist. To repair this, one could alternately define the value of a state as

$$v_\pi(s) \doteq \lim_{\gamma \to 1} \lim_{h \to \infty} \sum_{t=0}^{h} \gamma^t \left(\mathbb{E}_\pi[R_{t+1} | S_0 = s] - r(\pi) \right).$$

Under this definition, what are the values of states A and B?

1.7.2 A

Define

$$f(h) = \frac{1}{2h} \sum_{t=0}^{2h} \mathbb{1}\{t \equiv 0 \pmod 2\} = \frac{h+1}{2h}$$

then

$$\bar{R} = \lim_{h \to \infty} f(h/2) = \lim_{h \to \infty} f(h) = \frac{1}{2}.$$

Now to compute the differential state values we write

$$V(S; \gamma) = \lim_{h \to \infty} \sum_{t=0}^{h} \gamma^t \left(\mathbb{E}[R_{t+1} | S_0 = s] - \bar{R} \right)$$

then

$$\begin{aligned} V(A; \gamma) &= 1 - \bar{R} + \gamma V(A; \gamma) \\ V(A; \gamma) &= -\bar{R} + \gamma V(B; \gamma) \end{aligned}$$

so

$$V(A; \gamma) = \frac{1}{2}(1 - \gamma) - \gamma^2 V(A; \gamma)$$

and

$$\begin{aligned} V(A; \gamma) &= \frac{1}{2} \frac{1 - \gamma}{1 - \gamma^2} \\ &= \frac{1}{2(1 + \gamma)}. \end{aligned}$$

Finally, $V(A) = \lim_{\gamma \to 1} V(A; \gamma) = \frac{1}{4}$ and $V(B) = -\frac{1}{4}$.

1.8 Exercise 10.8

1.8.1 Q

The pseudocode in the box on page 251 updates $\bar{R}_{t+1}$ using δ_t as an error rather than simply $R_{t+1} - \bar{R}_{t+1}$. Both errors work, but using δ_t is better. To see why, consider the ring MRP of three states from Exercise 10.6. The estimate of the average reward should tend towards its true value of

$\frac{1}{3}$. Suppose it was already there and was held stuck there. What would the sequence of $R_{t+1} - \bar{R}_{t+1}$ errors be? What would the sequence of δ_t errors be (using (10.10))? Which error sequence would produce a more stable estimate of the average reward if the estimate were allowed to change in response to the errors? Why?

1.8.2 A

$\bar{R} = \frac{1}{3}$ fixed.

The sequence of errors from $R_t - \bar{R}_t$ starting in A would be

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

while the sequence of TD errors starting in A (taking differential values from Exercise 10.6) would be

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

which is clearly of much lower variance and would therefore give more stable updates. Once $\bar{R}$ gets to the correct value it never leaves.

1.9 Exercise 10.9

1.9.1 Q

In the differential semi-gradient n-step Sarsa algorithm, the step-size parameter on the average reward, β, needs to be quite small so that $\bar{R}$ becomes a good long-term estimate of the average reward. Unfortunately, $\bar{R}$ will then be biased by its initial value for many steps, which may make learning inefficient. Alternatively, one could use a sample average of the observed rewards for $\bar{R}$. That would initially adapt rapidly but in the long run would also adapt slowly. As the policy slowly changed, $\bar{R}$ would also change; the potential for such long-term non-stationarity makes sample-average methods ill-suited. In fact, the step-size parameter on the average reward is a perfect place to use the unbiased constant-step-size trick from Exercise 2.7. Describe the specific changes needed to the boxed algorithm for differential semi-gradient n-step Sarsa to use this trick.

1.9.2 A

We define a parameter β and seed a sequence u_n with $u_0 = 0$. The under the if statement where $\tau \geq 0$ we place the following:

$$u \leftarrow u + \beta(1 - u)$$
$$\bar{R} \leftarrow \bar{R} + \frac{\beta}{\mu}(R - \bar{R})$$