The University of Vermont

CS253A QR: Reinforcement Learning
Assignment №7

Ayat Ospanov

October 18, 2018

## Contents



# 1 Exercise 6.2

TD is better, as it updates as soon as it gets the data, i.e. at the next step. In the given example, we update our new time to the entrance to the highway after we get there. But MC doesn't update. It updates only when the state is terminal, i.e. we are at home. Thus, we need to make more steps before we update all values.

# 2 Exercise 6.3

It went straight to the left until the termination state. As values update as follows:

$$V(s) \leftarrow V(s) + \alpha[R + \gamma V(s') - V(s)]$$

we get:

$$V(A) = 0.5 + 0.1 * [0 + 0 - 0.5] = 0.45$$

# 3 Exercise 6.4

There is no need to check wider range of $\alpha$, as all the graphs will be between the graphs for $\alpha \to 0+$ and $\alpha \to 1-$. So in average, error for TD on the graphs would be lower, than for MC (at least at the beginning). There is no fixed $\alpha$ at which either algorithm would have performed significantly better, because we have $\alpha$'s close to 0 and 1 on the graph, and as we said before, all other $\alpha$'s are between them.

# 4 Exercise 6.6

## The first method. Linear system for Bellman equation for V(s)

As we have no actions, $\gamma = 1, p(s', r|s) = \frac{1}{2}$, we get:

$$v(s) = \sum_{s',r} \frac{1}{2}[r + v(s')]$$

Let's done $v(s)$ as $v_s$

$$\begin{cases} v_A = \frac{1}{2}[v_B] \\ v_B = \frac{1}{2}[v_A + v_C] \\ v_C = \frac{1}{2}[v_B + v_D] \\ v_D = \frac{1}{2}[v_C + v_E] \\ v_E = \frac{1}{2}[v_D + 1] \end{cases} \overset{5\to4,1\to2}{\Longrightarrow} \begin{cases} v_A = \frac{1}{2}v_B \\ v_B = \frac{2}{3}v_C \\ v_C = \frac{1}{2}[v_B + v_D] \\ v_D = \frac{1}{3}[2v_C + 1] \\ v_E = \frac{1}{2}[v_D + 1] \end{cases} \overset{4\to3}{\Longrightarrow} \begin{cases} v_A = \frac{1}{2}v_B \\ v_B = \frac{2}{3}v_C \\ v_C = \frac{1}{4}[3v_B + 1] \\ v_D = \frac{1}{3}[2v_C + 1] \\ v_E = \frac{1}{2}[v_D + 1] \end{cases}$$

From 2nd and 3rd equations we get $v_C = \frac{1}{2}$, and finally we get:

$$\begin{cases} v_A = \frac{1}{6} \\ v_B = \frac{1}{3} \\ v_C = \frac{1}{2} \\ v_D = \frac{2}{3} \\ v_E = \frac{5}{6} \end{cases} \Rightarrow \begin{cases} v_A = \frac{1}{6} \\ v_B = \frac{2}{6} \\ v_C = \frac{3}{6} \\ v_D = \frac{4}{6} \\ v_E = \frac{5}{6} \end{cases}$$

## The second method

Couldn't come up with the second, but it is around the probability theory.

# 5 Exercise 6.8

$$
\begin{aligned}
G_t - Q(S_t, A_t) &= R_{t+1} + \gamma G_{t+1} - Q(S_t, A_t) \pm \gamma Q(S_{t+1}, A_{t+1}) \\
&= \delta_t + \gamma(G_{t+1} - Q(S_{t+1}, A_{t+1})) = \\
&= \delta_t + \gamma\delta_{t+1} + \gamma^2(G_{t+2} - Q(S_{t+2}, A_{t+2})) = \\
&= \delta_t + \gamma\delta_{t+1} + \gamma^2\delta_{t+2} + ... = \\
&= \sum_{k=t}^{T-1} \gamma^{k-t}\delta_k
\end{aligned}
$$

# 6 Exercise 6.11

Q-learning is off-policy method, because the next step $a' = \max a'$ is drawn from another policy than our current next action from the generated episode.