Class Tutorial 9

In this tutorial we will consider the task of *evaluating the value of a policy* without having access to the model of the environment. We will solely assume access to sampled data and explore the TD(0), Monte-Carlo and TD(\(\lambda\)) algorithms.

1. The TD(0) Algorithm

Consider the following (stochastic shortest path) MDP:

![Diagram of an MDP with states a, b, c, d, e, f and transitions with rewards R=0, R=1, R=2 and w.p. probabilities 0.5 and 0.3.]

There is only one (trivial) policy here, which we denote by \(\pi\).

a. Compute the values \(V_{\pi}(s)\) for each state.

b. Consider running \(N\) trajectories \(Data = \{s_0^1, \ldots, s_T^1\}, \ldots, \{s_0^N, \ldots, s_T^N\}\) from the MDP using \(\pi\) and starting from \(s_0 = a\), for example:

\[
\begin{align*}
\{s_0^1, \ldots, s_T^1\} &= \{a, b, c, f, T\}, \\
\{s_0^2, \ldots, s_T^2\} &= \{a, d, e, T\}, \\
\{s_0^3, \ldots, s_T^3\} &= \{a, d, e, T\}, \\
&\ldots
\end{align*}
\]

Suggest an offline Monte-Carlo algorithm for estimating the values \(V_{\pi}(s)\) for each state, using \(Data\).

c. Consider running the TD(0) algorithm with the same data, starting from \(\hat{V}_{TD}(s) = 0\) for all states. Write down the execution of the algorithm for the first few iterations. Choose a step size \(\alpha_n = 1/\text{(no. of visits to } s_n)\).

d. Consider running the TD(0) algorithm again, but now assume that the values of states \(b, \ldots, f\) start from their true values, and do not change during the run of the algorithm. Show that \(\hat{V}_{TD}(a)\) converges to its true value.
Solution

a. \( V^\pi(s) = E^\pi \left[ \sum_{t=0}^{T} r(s_t) \mid s_0 = s \right] \), and \( V^\pi(s') = E^\pi \left[ r(s) + V^\pi(s') \right] \), therefore

\[
V^\pi(f) = 1, V^\pi(c) = 0.5, V^\pi(e) = 1, V^\pi(b) = 1.5, V^\pi(d) = 3, V^\pi(a) = 1 + (1.5 + 3) / 2.
\]

b. For each state \( s \), let \( D_s = \{ s, s_1, \ldots, s_{N_s} \} \) denote parts of the trajectories that start at \( s \), out of all the trajectories in \( D \) that path through \( s \). Then the MC estimate is

\[
\hat{V}_{MC}(s) = \frac{1}{N_s} \sum_{i=1}^{N_s} (r(s) + \ldots + r(s_{i_t})) .
\]

c. TD(0):

\[
\hat{V}_{TD}(s_a) := \hat{V}_{TD}(s_a) + \alpha_a \left( r(s_a) + \hat{V}_{TD}(s_{a+1}) - \hat{V}_{TD}(s_a) \right)
\]

Following the state sequences in the example:

\[
\begin{align*}
\hat{V}_{TD}(a) &= \hat{V}_{TD}(a) + \alpha_a \left( r(a) + \hat{V}_{TD}(b) - \hat{V}_{TD}(a) \right) = 1(1 + 0 - 0) \\
\hat{V}_{TD}(b) &= \hat{V}_{TD}(b) + \alpha_a \left( r(b) + \hat{V}_{TD}(c) - \hat{V}_{TD}(b) \right) = 1(1 + 0 - 0) \\
\hat{V}_{TD}(c) &= \hat{V}_{TD}(c) + \alpha_a \left( r(c) + \hat{V}_{TD}(f) - \hat{V}_{TD}(c) \right) = 1(0 + 0 - 0) \\
\hat{V}_{TD}(f) &= \hat{V}_{TD}(f) + \alpha_a \left( r(f) + \hat{V}_{TD}(T) - \hat{V}_{TD}(f) \right) = 1(1 + 0 - 0) \\
\hat{V}_{TD}(a) &= \hat{V}_{TD}(a) + \alpha_a \left( r(a) + \hat{V}_{TD}(d) - \hat{V}_{TD}(a) \right) = 1 + \frac{1}{2} (1 + 2 - 1) \\
&\text{...}
\end{align*}
\]

d. We have in this case

\[
\hat{V}_{TD}(a) = r(a) + \frac{1}{N_a} \sum_{i=1}^{N_a} V^\pi(s_a^i) = E^\pi \left[ r(a) + V^\pi(s_a^2) \right] = V^\pi(a)
\]

2. The TD(\( \lambda \)) Algorithm

Recall that in the TD(0) algorithm without function approximation the update for the value function is \( V_{t+1} \left( s_t \right) = V_t \left( s_t \right) + \alpha_t \left( r(s_t) + \gamma V_t \left( s_{t+1} \right) - V_t \left( s_t \right) \right) \), where the intuition behind it is that \( r(s_t) + \gamma V_t \left( s_{t+1} \right) \) is an estimate for \( V_t \left( s_t \right) \), and \( \delta_t = r(s_t) + \gamma V_t \left( s_{t+1} \right) - V_t \left( s_t \right) \) is thus the 1-step error term. One may similarly take \( r(s_t) + \gamma r(s_{t+1}) + \gamma^2 V_t \left( s_{t+2} \right) \) as an estimate for \( V_t \left( s_t \right) \), and define a 2-step error term \( \delta_2 = r(s_t) + \gamma r(s_{t+1}) + \gamma^2 V_t \left( s_{t+2} \right) - V_t \left( s_t \right) \). Similarly, we may define \( \delta_3, \delta_4, \ldots \).

The TD(\( \lambda \)) error is defined as a weighted average of \( \delta_1, \delta_2, \ldots \) as follows:
Learning and Planning in Dynamical Systems – 046194

\[ \delta^\lambda = (1 - \lambda) \sum_{i=1}^\infty \lambda^{i-1} \delta^i \]

a. For a state sequence of length 4: \( s_1, s_2, s_3, s_4 \) write down the TD(\( \lambda \)) errors explicitly for each state. Explain how a TD(\( \lambda \)) policy evaluation may be implemented.

One difficulty with the implementation above is that it seems difficult to implement as an online algorithm. However, a simple trick allows to stream-line the calculation as follows.

b. Show that for a long sequence of states \( s_1, s_2, s_3, \ldots \) the error term \( \delta^\lambda(s_1) \) may be written as a combination of \( \delta^0(s_1), \delta^1(s_2), \ldots \)

c. Propose an online implementation of the TD(\( \lambda \)) algorithm.

**Solution**

a. 

\[
\delta^\lambda(s_1) = \left(1 - \lambda\right)\left( r(s_1) + \gamma V(s_2) - V(s_1) \right) \\
+ \lambda \left(1 - \lambda\right)\left( r(s_1) + \gamma r(s_2) + \gamma^2 V(s_3) - V(s_1) \right) \\
+ \lambda^2 \left(1 - \lambda\right)\left( r(s_1) + \gamma r(s_2) + \gamma^2 r(s_3) + \gamma^3 V(s_4) - V(s_1) \right)
\]

\[
\delta^\lambda(s_2) = \left(1 - \lambda\right)\left( r(s_2) + \gamma V(s_3) - V(s_2) \right) \\
+ \lambda \left(1 - \lambda\right)\left( r(s_2) + \gamma r(s_3) + \gamma^2 V(s_4) - V(s_2) \right)
\]

\[
\delta^\lambda(s_3) = \left(1 - \lambda\right)\left( r(s_3) + \gamma V(s_4) - V(s_3) \right)
\]

b. note that

\[
\delta^\lambda(s_1) = -V(s_1) + \left(1 - \lambda\right)\left( r(s_1) + \gamma V(s_2) \right) \\
+ \lambda \left(1 - \lambda\right)\left( r(s_1) + \gamma r(s_2) + \gamma^2 V(s_3) \right) \\
+ \lambda^2 \left(1 - \lambda\right)\left( r(s_1) + \gamma r(s_2) + \gamma^2 r(s_3) + \gamma^3 V(s_4) \right) \\
+ \ldots
\]

By taking out \( r(s_1) \) we see that its coefficients sum up to 1. Doing this for the other terms gives:
\[ \delta^2(s) = -V(s) + (\gamma \lambda)^0 (r(s) + \gamma V(s) - \gamma \lambda V(s)) + (\gamma \lambda)^1 (r(s) + \gamma V(s) - \gamma \lambda V(s)) + \ldots = \sum_{t=1}^{\infty} (\gamma \lambda)^{t-1} \delta_0(s) \]

c.
\[ V_{e+1}(s) = V_e(s) + \alpha_t \delta e_t(s) \quad \forall s \]
\[ \delta_t = r(s) + \gamma V(s_{e+1}) - V_e(s) \]
\[ e_t(s) = \gamma \lambda e_{t-1}(s) + \mathbb{1}_{s_{e+1} = s} \quad \forall s \]