Reinforcement Learning

Lecture 4

Lecturer: Haim Permuter *Scribe: Tal Edvabsky*

I. Temporal-Difference Learning

In the last lecture, we covered Monte-Carlo (MC) Learning. Using MC, we can learn directly from raw experience without knowing the environment's dynamics. In this lecture, we will discuss a similar method called Temporal-Difference (TD) Learning, which is undoubtedly one of the most important concepts in Reinforcement Learning. Similar to MC in the sense that it does not require a model, it is a model-free algorithm that also resembles Dynamic-Programming (DP) in the sense that it updates estimates based on other learned estimates without waiting for the final outcome. Its similarity to DP is the largest difference between TD learning and MC learning. In this lecture, we will focus only on the policy evaluation, i.e., the "prediction problem".

II. Incremental Average

In this section we will develop a simple but useful mathematical equality between the current mean μ_k and the previous mean μ_{k-1}:

$$\begin{aligned}\mu_k &= \frac{1}{k}\sum_{t=1}^{k} x_t \\ &= \frac{1}{k}\left(\sum_{t=1}^{k-1} x_t + x_k\right) \\ &= \frac{1}{k}\left((k-1)\,\mu_{k-1} + x_k\right) \\ &= \mu_{k-1} + \frac{1}{k}\left(x_k - \mu_{k-1}\right).\end{aligned}$$

The final result gives:

$$\mu_k = \mu_{k-1} + \frac{1}{k}\left(x_k - \mu_{k-1}\right). \tag{1}$$

III. TD(0) PREDICTION

Let's start with a reminder of the Incremental MC updates:

$$N(S_t) \leftarrow N(S_t) + 1, \tag{2}$$

$$V(S_t) \leftarrow V(S_t) + \frac{1}{N(S_t)}(G_t - V(S_t)). \tag{3}$$

Note that Eq. (3) is dependent on the formula developed in Eq. (1). In a non-stationary problem, we can substitute the term $\frac{1}{N(S_t)}$ with α and Eq. (3) becomes:

$$V(S_t) \leftarrow V(S_t) + \alpha(G_t - V(S_t)), \tag{4}$$

where

$$G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}. \tag{5}$$

In $TD(0)$, we replace the term G_t, which is the actual return, with the estimated return $R_{t+1} + \gamma V(S_{t+1})$ to get:

$$V(S_t) \leftarrow V(S_t) + \alpha(R_{t+1} + \gamma V(S_{t+1}) - V(S_t)), \tag{6}$$

where $R_{t+1} + \gamma V(S_{t+1})$ is called the "TD Target" and $\delta_t = R_{t+1} + \gamma V(S_{t+1}) - V(S_t)$ is called the "TD error". The equations above combine to create the famous $TD(0)$ algorithm:

Algorithm 1 $TD(0)$ for estimating v_π

Input: the policy π to be evaluated.

Initialize: $V(s)$ arbitrary.

repeat

Initialize S

For each step of episode:

$A \leftarrow$ action given by π for S

Take action A and observe R, S'

$V(S_t) \leftarrow V(S_t) + \alpha[R_{t+1} + \gamma V(S_{t+1}) - V(S_t)]$

$S \leftarrow S'$

until S is terminal

IV. Advantages and Disadvantages of $TD(0)$ learning

One can observe from Eq. (6) that $TD(0)$ bootstraps. Bootstrapping means that the algorithm does not need to wait until the end of an episode to evaluate the value function; it estimates $v_\pi(s)$ one step ahead. Let's summarize and compare the advantages and disadvantages of $TD(0)$ and MC:

- TD can learn every step while MC must wait until the end of the episode (online learning).
- In contrast to MC, TD can work in non-terminating environments.
- The TD Target $R_{t+1} + \gamma V(S_{t+1})$ is a biased estimate of $v_\pi(s)$, while in MC, the return is an unbiased estimate.
- TD Target has much lower variance than the return G_t used in MC Learning.

To understand the differences between MC and $TD(0)$ more clearly and to obtain better intuition about the performance of this method, we will use a simple example. Let's take an MDP with only 2 states: A,B. There is no discount factor ($\gamma = 1$), and we use 8 episodes of experience as follows:

1) A,0,B,0.
2) B,1.
3) B,1.
4) B,1.
5) B,1.
6) B,1.
7) B,0.

What are $V(A)$ and $V(B)$?

Solution:

- The MC algorithm converges to a solution with minimum mean-squared error (MMSE):

$$\sum_{k=1}^{K} \sum_{t=1}^{T_k} (G_t^k - V(s_t^k))^2. \tag{7}$$

  In the example above, using MC we get: $V(A) = 0$.

- The $TD(0)$ algorithm converges to a solution of a maximum likelihood Markov process, which means that it fits the data:

$$\Pr(S_{t+1} = s' | S_t = s, A_t = a) = \frac{1}{N(s,a)} \sum_{k=1}^{K} \sum_{t=1}^{T_s} \mathbf{1}(s_t^k, a_t^k, s_{t+1}^k = s, a, s'), \quad (8)$$

$$r(s,a) = E[R_t | S_t = s, A_t = a] = \frac{1}{N(s,a)} \sum_{k=1}^{K} \sum_{t=1}^{T_s} \mathbf{1}(s_t^k, a_t^k = s, a) r_t^k. \quad (9)$$

  In the example above, using $TD(0)$ we get: $V(A) = \frac{6}{8}$. In both algorithms, $V(B) = \frac{6}{8}$.

V. n-STEP TD PREDICTION

Until now, we discussed about approximating the TD target by looking one step into the future. What about looking 2 steps into the future? Or 3 steps? Let's see how the TD target changes as a function of the number of steps:

- **1 step:** $G_t^{(1)} = R_{t+1} + \gamma V(S_{t+1})$
- **2 steps:** $G_t^{(2)} = R_{t+1} + \gamma R_{t+2} + \gamma^2 V(S_{t+2})$
- **3 steps:** $G_t^{(3)} = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 V(S_{t+3})$
- **n steps:** $G_t^{(n)} = R_{t+1} + \gamma R_{t+2} + ... + \gamma^{n-1} R_{t+n} + \gamma^n V(S_{t+n})$

One can observe that when $n \to \infty$, TD becomes MC. In conclusion, the n-step TD learning algorithm is:

Algorithm 2 $TD(0)$ for estimating v_π with n-step TD prediction

Input: the policy π to be evaluated.

Initialize: $V(s)$ arbitrary.

repeat

Initialize S

For each step of episode:

$A \leftarrow$ action given by π for S

Take action A and observe R, S'

$V(S_t) \leftarrow V(S_t) + \alpha[G_t^{(n)} - V(S_t)]$

$S \leftarrow S'$

until S is terminal

VI. TD(λ)

What is the parameter λ in $TD(\lambda)$? Until now, λ has always been equal to zero. In light of the n-step prediction in Algorithm V, one can deduce another possibility. Perhaps averaging the n-step returns over different $n's$ can give a better prediction and a faster convergence to the value function. For example, one can take $\frac{1}{2}G^{(3)} + \frac{1}{2}G^{(5)}$ as a possible estimated return for all states $s \in \mathscr{S}$. Likewise, we can combine information from **two** different time steps, and indeed, we can efficiently combine information from **all** time steps as follows:

$$G_t^\lambda = (1-\lambda)\sum_{n=1}^{\infty} \lambda^{n-1} G_t^{(n)}. \tag{10}$$

The complete algorithm, which is also called the forward view $TD(\lambda)$, is:

Algorithm 3 $TD(\lambda)$ - forward view for estimating v_π

Input: the policy π to be evaluated.

Initialize: $V(s)$ arbitrary.

repeat

Initialize S

For each step of episode:

$A \leftarrow$ action given by π for S

Take action A and observe R, S'

$V(S_t) \leftarrow V(S_t) + \alpha(G_t^\lambda - V(S_t))$

$S \leftarrow S'$

until S is terminal

This algorithm is termed the forward view $TD(\lambda)$ because it looks into the future to compute G_t^λ. Similar to MC, it only updates at the end of an episode.

VII. Backward View $TD(\lambda)$ and Eligibility Traces

While the forward view case provides one with a good understanding of how $TD(\lambda)$ works, the backward view is the more practical of the two approaches.In contrast to the

forward view, it updates online, every step, from incomplete episodes by using eligibility traces.

A. Eligibility Traces

Eligibility traces are one of the basic mechanisms of RL. For every state $s \in \mathscr{S}$, we keep a scalar $E_t(s)$ that follows:

$$E_0(s) = 0, \tag{11}$$

$$E_t(s) = \gamma\lambda E_{t-1}(s) + \mathbf{1}(S_t = s). \tag{12}$$

One can infer from Eq. (12) that:

- $\gamma\lambda E_{t-1}(s)$ assigns credit to the most frequent states.
- $\mathbf{1}(S_t = s)$ assigns credit to the most recent states.

In proportion to TD error δ_t and eligibility trace $E_t(s)$, we obtain the value function update:

$$\delta_t = R_{t+1} + \gamma V(S_{t+1}) - V(S_t), \tag{13}$$

$$V(s) \leftarrow V(s) + \alpha\delta_t E_t(s). \tag{14}$$

To summarize, the backward view of $TD(\lambda)$ is as follows:

Algorithm 4 $TD(\lambda)$ - backward view for estimating v_π

Input: the policy π to be evaluated.

Initialize: $V(s)$ arbitrary and $E(s) = 0$ for all $s \in \mathscr{S}$.

repeat

Initialize S

For each step of episode:

$A \leftarrow$ action given by π for S

Take action A and observe R, S'

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

$E_t(s) \leftarrow E_t(s) + 1$

for all $s \in \mathscr{S}$:

$V(s) \leftarrow V(s) + \alpha \delta_t E_t(s)$

$E_t(s) \leftarrow \gamma \lambda E_t(s)$

$S \leftarrow S'$

until S is terminal

Now let's consider two special cases of $TD(\lambda)$:

- $\lambda = 0$: Only the current state is updated. The "frequent" property of eligibility traces vanishes leaving only the "recent" property stays. The update becomes equivalent to the $TD(0)$ update:

$$E_t(s) = \mathbf{1}(S_t = s), \tag{15}$$

$$V(s) \leftarrow V(s) + \alpha \delta_t. \tag{16}$$

- $\lambda = 1$: Credit is postponed until the end of an episode. Over the course of an episode, the total update for $TD(1)$ is the same as the total update for MC.