Model Free Prediction : Multi-Step TD Methods

Easwar Subramanian

TCS Innovation Labs, Hyderabad

Email : easwar.subramanian@tcs.com / cs5500.2020@iith.ac.in

September 23, 2021

Administrivia

- Assignment 1 is due on Monday (27th Jan, 2021). Please submit them on time.
- Assignment 2 will be posted in the first week of October.
- Mid-term exam will likely be in the thrid week of October – Details to follow soon
- Project proposal deadline is September 30th, 2021

Overview

1. Review
2. Multi-Step Temporal Difference

Review

Temporal Difference : Key Idea

$$
\begin{aligned}
V^{\pi}(s) &\overset{\text{def}}{=} \mathbb{E}_{\pi}(G_t|s_t = s) = \mathbb{E}_{\pi}\left(\sum_{k=0}^{\infty} \gamma^k r_{t+k+1}|s_t = s\right) \\
&= \mathbb{E}_{\pi}\left[r_{t+1} + \gamma V^{\pi}(s_{t+1})|s_t = s\right]
\end{aligned}
$$

- Estimate expectation from experience using the recursive decomposition formulation of the value function

General Form of Update Rule

The general form for the update rule that is present in the incremental calculation is,

$$\text{New Estimate} \leftarrow \text{Old Estimate} + \text{Learning Rate(Target - Old Estimate)}$$

- The expression (Target - Old Estimate) is an error of the estimate
- The error is reduced by taking steps towards the ”Target”
- The target is persumed to indicate a desirable direction to move
- In the incremental calculation of mean, the term x_{k+1} is the target

One-Step TD

- We wish to approximate

$$V^\pi(s) = \mathbb{E}_\pi\left[r_{t+1} + \gamma V^\pi(s_{t+1}) | s_t = s\right]$$

- Approximate the expectation by a sample mean
  - ★ If the *transition* (s_t, r_{t+1}, s_{t+1}) is observed at time t under π, then

$$V(s_t) \leftarrow V(s_t) \; + \; \alpha_t[r_{t+1} + \gamma V(s_{t+1}) - V(s_t)]$$

  - ★ Samples come from different visits to the state s, either from same or different trajectories
  - ★ Compute the sample mean incrementally

One-Step TD : TD(0) Algorithm

Algorithm TD(0) : Algorithm

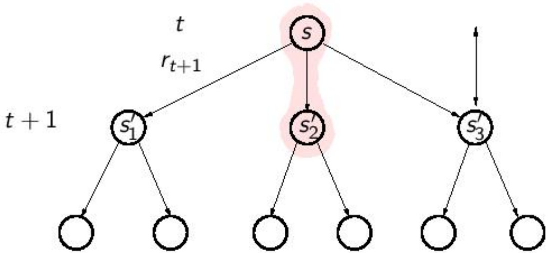
```
Initialize $V(s)$ arbitrarily (say, $V(s) = 0 \quad \forall s \in \mathcal{S}$);
for $k = 1, 2, \cdots, K$ do
    Let $s$ be a start state for episode $k$
    for For each step in the $k$-th episode do
        Take action $a$ recommended by policy $\pi$ from state $s$
        Collect reward $r$ and reach next state $s'$
        Perform the following TD update
            $V(s) = V(s) + \alpha_{N(s)}[r + \gamma V(s') - V(s)]$
        Assign $s \leftarrow s'$
    end for
end for
```

Convergence of TD Algorithms

- For any fixed policy π, the TD(0) algorithm described above converges (asymptotically) to V^{π} under some conditions on the choice of α (Robbins Monroe Condition)
  - ★ $\sum \alpha_t = \infty$
  - ★ $\sum \alpha_t^2 < \infty$

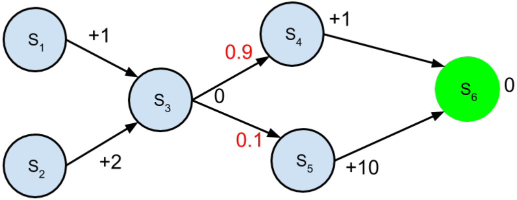
- *Generally*, TD methods have usually been found to converge faster than MC methods on certain class of tasks

TD Algorithms: A Schematic View



- Uses experience without model like MC
- Bootstraps like DP
- Can work with partial sequences
- Suited for online learning

TD vs MC : Example



(1) $s_1 \xrightarrow{1} s_3 \xrightarrow{0} s_4 \xrightarrow{1} s_6$

(2) $s_1 \xrightarrow{1} s_3 \xrightarrow{0} s_5 \xrightarrow{10} s_6$

(3) $s_1 \xrightarrow{1} s_3 \xrightarrow{0} s_4 \xrightarrow{1} s_6$

(4) $s_1 \xrightarrow{1} s_3 \xrightarrow{0} s_4 \xrightarrow{1} s_6$

(5) $s_2 \xrightarrow{2} s_3 \xrightarrow{0} s_5 \xrightarrow{10} s_6$

Slide Credit: Abir Das: : IIT-KG

TD vs MC : Example

- True value of each state is given by $V(s_6) = 0, V(s_5) = 10, V(s_4) = 1, V(s_3) = 1.9, V(s_2) = 3.9$ and $V(s_1) = 2.9$
- Evaluate $V(s_1)$ and $V(s_2)$ using MC $V(s_1) = 4.25$ and $V(s_2) = 12$
- Evaluate $V(s_1)$ and $V(s_2)$ using TD (Evaluating left to right)
  - ★ First trajectory $(s_1 \xrightarrow{1} s_3 \xrightarrow{0} s_4 \xrightarrow{1} s_6)$
    $V(s_1) = 1;\ V(s_3) = 0;\ V(s_4) = 1;\ V(s_6) = 0$
  - ★ Second trajectory $(s_1 \xrightarrow{1} s_3 \xrightarrow{0} s_5 \xrightarrow{10} s_6)$
    $V(s_1) = 1;\ V(s_3) = 0;\ V(s_5) = 10;\ V(s_6) = 0$
  - ★ Third trajectory $(s_1 \xrightarrow{1} s_3 \xrightarrow{0} s_4 \xrightarrow{1} s_6)$
    $V(s_1) = 1;\ V(s_3) = 0.33;\ V(s_4) = 1;\ V(s_6) = 0$
  - ★ Fourth trajectory $(s_1 \xrightarrow{1} s_3 \xrightarrow{0} s_4 \xrightarrow{1} s_6)$
    $V(s_1) = 1.08;\ V(s_3) = 0.5;\ V(s_4) = 1;\ V(s_6) = 0$
  - ★ Fifth trajectory $(s_2 \xrightarrow{2} s_3 \xrightarrow{0} s_5 \xrightarrow{10} s_6)$
    $V(s_2) = 2.5;\ V(s_3) = 2.4;\ V(s_5) = 10;\ V(s_6) = 0$

TD Vs MC : Example

Evaluate $V(s_1)$ and $V(s_2)$ using TD (Evaluating right to left)

- First trajectory $(s_1 \xrightarrow{1} s_3 \xrightarrow{0} s_4 \xrightarrow{1} s_6)$
  $V(s_6) = 0$; $V(s_4) = 1$; $V(s_3) = 1$; $V(s_1) = 2$
- Second trajectory $(s_1 \xrightarrow{1} s_3 \xrightarrow{0} s_5 \xrightarrow{10} s_6)$
  $V(s_6) = 0$; $V(s_5) = 10$; $V(s_3) = 5.5$; $V(s_1) = 4.25$
- Third trajectory $(s_1 \xrightarrow{1} s_3 \xrightarrow{0} s_4 \xrightarrow{1} s_6)$
  $V(s_6) = 0$; $V(s_4) = 1$; $V(s_3) = 4$; $V(s_1) = 4.5$
- Fourth trajectory $(s_1 \xrightarrow{1} s_3 \xrightarrow{0} s_4 \xrightarrow{1} s_6)$
  $V(s_6) = 0$; $V(s_4) = 1$; $V(s_3) = 3.25$; $V(s_1) = 4.43$
- Fifth trajectory $(s_2 \xrightarrow{2} s_3 \xrightarrow{0} s_5 \xrightarrow{10} s_6)$
  $V(s_6) = 0$; $V(s_5) = 10$; $V(s_3) = 4.6$; $V(s_2) = 6.6$

Takeaway

- **TD evaluation depend on order of experiences considered**

On Bias and Variance of MC and TD Estimators

Consider the following expression for V^π

$$V^\pi(s) \overset{\text{def}}{=} \mathbb{E}_\pi(G_t | s_t = s) = \mathbb{E}_\pi\left(\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s\right)$$

Question :

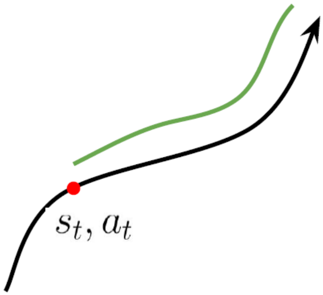
Which terms in the above expression creates uncertainity in evaluation of V^π ?

Answer : Immediate and Future rewards !

—

- The randomness in rewards are due to stochasaticity of the policy and environment

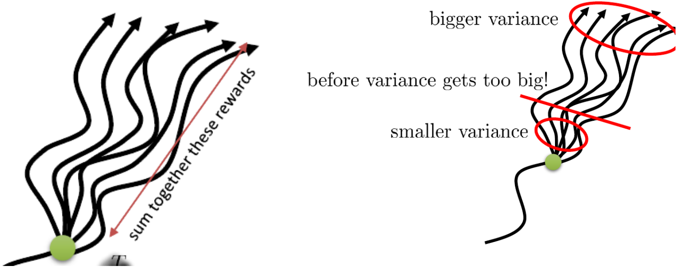
On Bias and Variance of MC and TD Estimators



The rewards are counted along the green curve and hence the green curve represents the summation in the defintion of V^{π}

Figure Source: Jie-Han-Chen:SlideShare

Bias-Variance Tradeoff in MC and TD Estimators



- ▶ In the MC method we need to wait till the end of the trajectory to make an update; It means lots of future rewards need to be summed. This makes the estimate have high variance
- ▶ In the TD, we cut off the path only until next state (in one step TD) and hence the estimate has low variance. Because of bootstrapping, the TD estimate has high bias

Figure Source: UCB : Sergey Levine

Properties of Different Policy Evaluation Algorithms

	DP Algorithms	MC Algorithms	TD Algorithms
Model Free	No	Yes	Yes
Non Episodic Domains	Yes	No	Yes
Non Markovian Domains	No	Yes	No
Bias	Not Applicable	Unbiased	Some Bias
Variance	Not Applicable	High Variance	Low Variance

Multi-Step Temporal Difference

Multi-step TD

- One-step TD

$$\begin{aligned} V^\pi(s) &= \mathbb{E}_\pi\left[r_{t+1} + \gamma V^\pi(s_{t+1}) | s_t = s\right] \\ V(s_t) &\leftarrow V(s_t) + \alpha_t[r_{t+1} + \gamma V(s_{t+1}) - V(s_t)] \end{aligned}$$

- Two-step TD

$$\begin{aligned} V^\pi(s) &= \mathbb{E}_\pi\left[r_{t+1} + \gamma r_{t+2} + \gamma^2 V^\pi(s_{t+2}) | s_t = s\right] \\ V(s_t) &\leftarrow V(s_t) + \alpha_t[r_{t+1} + \gamma r_{t+2} + \gamma^2 V(s_{t+2}) - V(s_t)] \end{aligned}$$

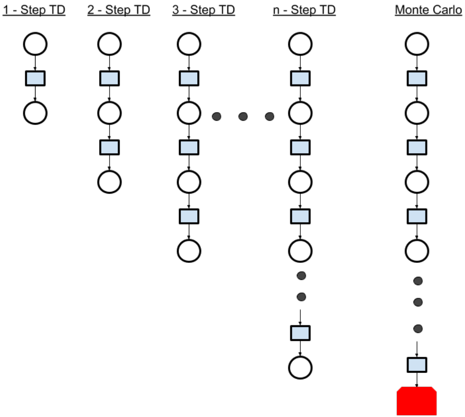
- More generally, define the n-step return

$$G_t^{(n)} \overset{\text{def}}{=} r_{t+1} + \gamma r_{t+2} + \cdots + \gamma^{n-1} r_{t+n} + \gamma^n V(s_{t+n})$$

- n-step TD

$$V(s_t) \leftarrow V(s_t) + \alpha_t[G_t^{(n)} - V(s_t)]$$

Why Multi-Step TD ?



- Multi-step TD methods tend to have less bias compared to 1-step TD method

Figure Source: Sutton and Barto

λ-Return

- What if we average some or all the n-step returns?

Example : Target could be

$$\frac{1}{2}G_t^{(2)} + \frac{1}{2}G_t^{(4)}$$

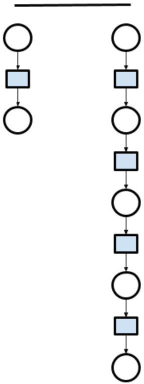


Figure Source: Sutton and Barto

λ-Return

- Any set of returns can be averaged, even an infinite set, as long as the weights on the component returns are positive and sum to 1
- Choose $\lambda \in [0, 1]$, and define the λ-return at time t as

$$G_t^\lambda \overset{\text{def}}{=} (1-\lambda) \sum_{n=1}^{\infty} \lambda^{n-1} G_t^{(n)}$$

- λ-return algorithm: use λ-return as the target

$$V(s_t) \leftarrow V(s_t) + \alpha_t [G_t^\lambda - V(s_t)]$$

- If the episode ends at $T > t$, define $G_t^{(n)}$ to be G_t for all $n > T - t$. Then

$$G_t^\lambda = (1-\lambda) \sum_{n=1}^{T-t} \lambda^{n-1} G_t^{(n)} + \lambda^{T-t} G_t$$

- $\lambda = 0$ gives 1-step TD, $\lambda = 1$ gives MC update

λ-Return

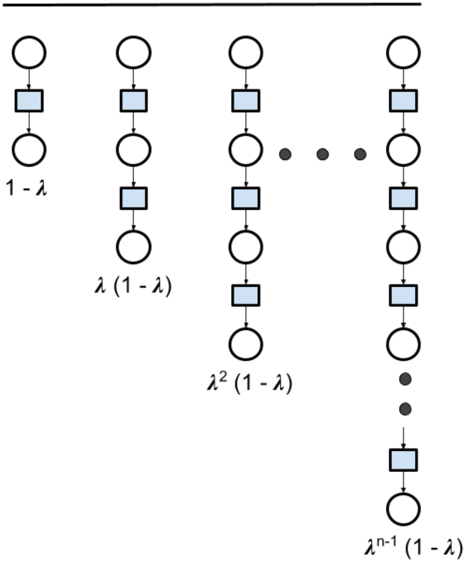
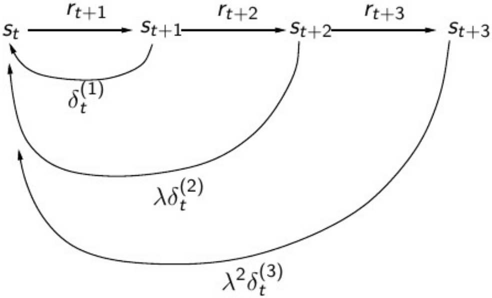


Figure Source: Sutton and Barto

Forward View

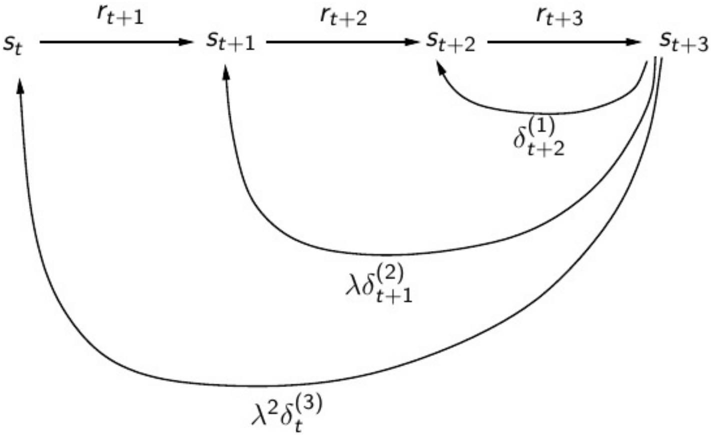
$$G_t^\lambda - V(s_t) = (1-\lambda)\sum_{n=1}^{\infty}\lambda^{n-1}[G_t^{(n)} - V(s_t)] = (1-\lambda)\sum_{n=1}^{\infty}\lambda^{n-1}\delta_t^{(n)}$$



- Not suitable for online implementation;

Figure Source: Sutton and Barto

A Possible Online Implementation



- Requires storing all rewards from the episode

Figure Source: Sutton and Barto

A Rearrangement of n-step TD Errors

$$\begin{aligned}
\delta_t^{(n)} &\overset{\text{def}}{=} r_{t+1} + \gamma r_{t+2} + \cdots + \gamma^{n-1} r_{t+n} + \gamma^n V(s_{t+n}) - V(s_t) \\
&= \quad \gamma^0[r_{t+1} + \gamma V(s_{t+1}) - V(s_t)] \\
&\quad + \gamma^1[r_{t+2} + \gamma V(s_{t+2}) - V(s_{t+1})] \\
&\vdots \\
&\quad + \gamma^{n-1}[r_{t+n} + \gamma V(s_{t+n}) - V(s_{t+n-1})] \\
&= \sum_{i=t}^{t+n-1} \gamma^{i-t} \delta_i^{(1)}
\end{aligned}$$

$$\begin{aligned}
G_t^\lambda - V(s_t) &= (1-\lambda) \sum_{n=1}^{\infty} \lambda^{n-1} \delta_t^{(n)} = (1-\lambda) \sum_{n=1}^{\infty} \lambda^{n-1} \sum_{i=t}^{t+n-1} \gamma^{i-t} \delta_i^{(1)} \\
&= \sum_{i=0}^{\infty} (\lambda\gamma)^i \delta_{t+i}^{(1)}
\end{aligned}$$

Online Implementation

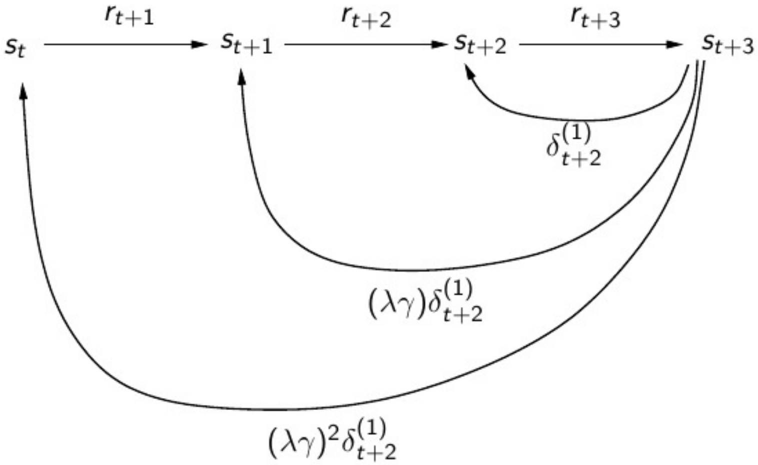


Figure Source: Sutton and Barto

The latest transition is used to update all previously encountered states

Eligibility Traces

- The eligibility trace of a state $s \in \mathcal{S}$ at time t is defined recursively by

$$\begin{aligned} e_0(s) &= 0 \\ e_t(s) &= \begin{cases} (\lambda\gamma)e_{t-1}(s), & s_t \neq s \\ (\lambda\gamma)e_{t-1}(s) + 1, & s_t = s \end{cases} \end{aligned}$$

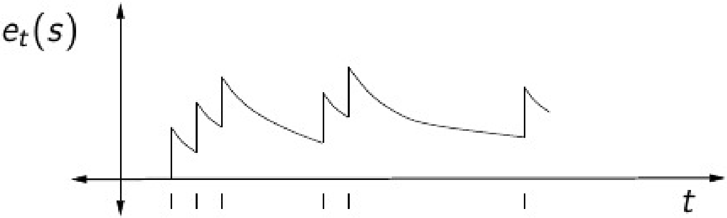


Figure Source: Sutton and Barto

Algorithm : TD(λ)

Algorithm TD(λ) : Algorithm

```
Initialize $e(s) = 0$ for all $s$, $V(s)$ arbitrarily
for For each episode do
    Let $s$ be a start state for episode $k$
    for For each step of the episode do
        Take action $a$ recommended by policy $\pi$ from state $s$
        Collect reward $r$ and reach next state $s'$
        Form the one-step TD error $\delta \leftarrow r + \gamma V(s') - V(s)$
        Increment eligibility trace of state $s$, $e(s) \leftarrow e(s) + 1$
        for For all states $S \in \mathcal{S}$ do
            Update $V(S)$: $V(S) \leftarrow V(S) + \alpha e(S) \delta$
            Update eligibility trace: $e(S) \leftarrow \lambda \gamma e(S)$
        end for
        Move to next state: $s \leftarrow s'$
    end for
end for
```