Model Free Prediction : Temporal Difference Methods

Easwar Subramanian

TCS Innovation Labs, Hyderabad

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

September 16, 2021

Overview

1. Review

2. Model Free Prediction : Temporal Difference

Review

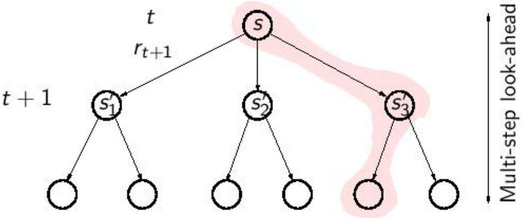
Model Free Methods : Key Idea

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

How can we estimate the expectations?
Use samples!

- Monte Carlo methods estimate $V^\pi(s)$ by accumulating rewards along different trajectories of π starting from state s
- By the law of large numbers $V(s) \rightarrow V^\pi(s)$ as number of episodes increases

Monte Carlo Algorithms: A Schematic View



- Uses experience, rather than model
- Uses only experience; does not bootstrap
- Needs complete sequences; suitable only for episodic tasks
- Suited for off-line learning
- Time required for one estimate does not depend on total number of states
- Can be used in non-Markovian setting as well

Model Free Prediction : Temporal Difference

Temporal Difference : Key Idea

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

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

Incremental Calculation of Mean

$$
\begin{aligned}
\mu_{k+1} &\overset{\text{def}}{=} \frac{1}{k+1}\sum_{i=1}^{k+1} x_i \\
&= \frac{1}{k+1}\sum_{i=1}^{k} x_i + \frac{1}{k+1}x_{k+1} \\
&= \frac{k}{k+1}\left(\frac{1}{k}\sum_{i=1}^{k} x_i\right) + \frac{1}{k+1}x_{k+1} \\
&= \frac{k}{k+1}\mu_k + \frac{1}{k+1}x_{k+1} \\
&= \mu_k + \frac{1}{k+1}(x_{k+1} - \mu_k)
\end{aligned}
$$

$$
\text{Update} = \text{learning rate} \times (\text{Target} - \text{Previous Value})
$$

General Form of Update Rule

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

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

- 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 π from state s
    Collect reward r and reach next state s'
    Perform the following TD update

$$V(s) = V(s) + \alpha[r + \gamma V(s') - V(s)]$$

    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

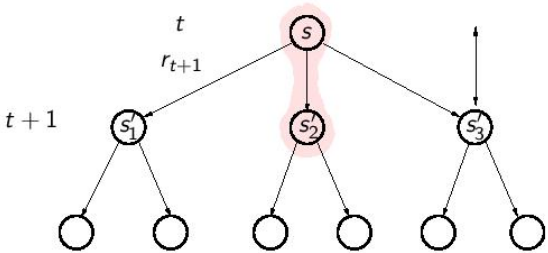
Connections between MC Error and TD Error

- The term $\delta_t = [r_{t+1} + \gamma V(s_{t+1}) - V(s_t)]$ is called the (one step) **TD error**
- The term $G_t - V(s_t)$ is called the MC error
- If a trajectory has T time steps then

$$G_t - V(s_t) = \sum_{k=0}^{T-t-1} \gamma^k \delta_{t+k}$$

$$(\text{Exercise}: \text{ Prove it !})$$

TD Algorithms: A Schematic View

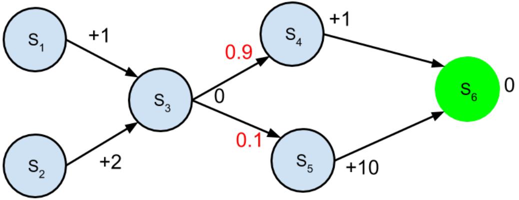


- Uses experience without model like MC
- Bootstraps like DP
- Can work with partial sequences
- Suited for online learning

TD vs MC: An Example

- Consider a Markov reward process with two states A and B
- You are given a dataset containing the following state-reward trajectories
  1) (A,0),(B,0) 2) (B,1) 3) (B,1) 4) (B,1) 5) (B,1) 6) (B,1) 7) (B,1) 8) (B,0)
- Estimate $V(A)$ using FV MC and TD
- $V(B)$ is $\frac{3}{4}$ can be easily read from the trajectory listing.
- $V(A)$ is 0 in MC Method and $\frac{3}{4}$ using TD

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

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
  - ★ 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 $(s_2 \xrightarrow{2} s_3 \xrightarrow{0} s_5 \xrightarrow{10} s_6)$
    $V(s_6) = 0; V(s_5) = 10; V(s_3) = 4.6; V(s_2) = 6.6$

Schematic View of Various Algorithms

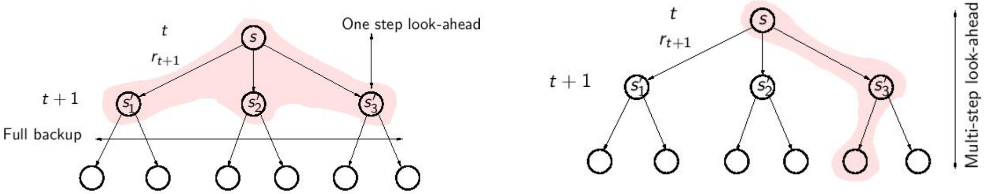
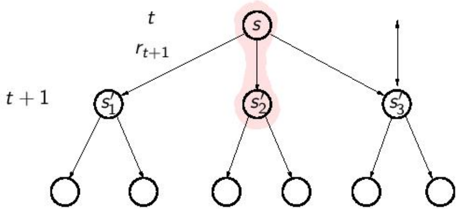


Figure: DP Algorithm and MC Algorithm



Bias/Variance Trade-Off in TD Algorithms

Monte Carlo Algorithms

- No Bias
  - ★ Sample average is an unbiased estimate of the expectation
- High Variance
  - ★ Return function of multi-step sequence of random actions, states and rewards

Temporal Difference Algorithms

- Some Bias
  - ★ TD target $r_{t+1} + \gamma V(s_{t+1})$ is a biased estimate of $V(s)$
- Low Variance
  - ★ TD target only has one random action, reward and next state

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