Chapter 6 - Temporal-Difference Learning: Jan 3, 2022

Temporal-Difference Learning combines Monte Carlo and dynamic programming ideas.

↳ Like MC methods, TD methods learn directly from experience w/t a model of the environment.

↳ Like DP, TD updates estimates based, in part, on other estimates w/t waiting for a final outcome (bootstrapping).

6.1 - TD Prediction:

Every-visit Monte Carlo:

$$V(S_t) \leftarrow V(S_t) + \alpha[G_t - V(S_t)]$$

✡ Must wait for episode to end to get G_t.

TD Method:

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

✡ TD methods can determine the increment to $V(S_t)$ after each time step.

✡ This is a special case of TD(λ) called TD(0) or one-step TD.

TD(0) Algorithm Pseudocode!

Tabular TD(0) for estimating v_π

Input: the policy π to be evaluated
Algorithm parameter: step size $\alpha \in (0, 1]$
Initialize $V(s)$, for all $s \in \mathcal{S}^+$, arbitrarily except that $V(terminal) = 0$

Loop for each episode:
  Initialize S
  Loop for each step of episode:
    $A \leftarrow$ action given by π for S
    Take action A, observe R, S'
    $V(S) \leftarrow V(S) + \alpha[R + \gamma V(S') - V(S)]$
    $S \leftarrow S'$
  until S is terminal

※ This is a bootstrapping method.

↳ $V_\pi(s) \equiv E_\pi[G_t | S_t = s]$

$= E_\pi[R_{t+1} + \gamma G_{t+1} | S_t = s]$

$= E_\pi[R_{t+1} + \gamma V_\pi(S_{t+1}) | S_t = s]$

TD Error!

$$\delta_t \equiv R_{t+1} + \gamma V(S_{t+1}) - V(S_t)$$ ※ TD Error

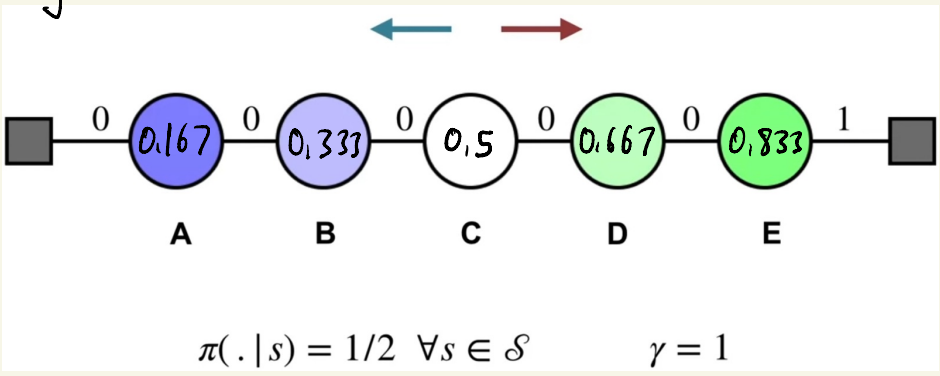
↳ From $V(S) \leftarrow V(S) + \alpha[R + \gamma V(S') - V(S)]$

6.2 - Advantages of TD Prediction Methods:

1. Do not require a model of the environment like DP methods.
2. Naturally implemented in an online, fully-incremental fashion. No need to wait until the end of an episode, like w/ Monte Carlo methods.
3. TD methods usually converge to ideal policies faster than MC methods.

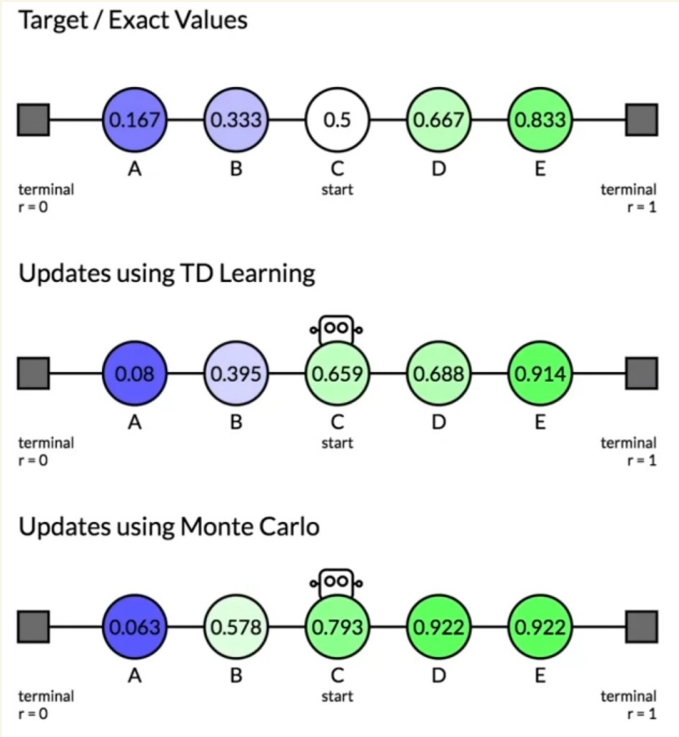
TD vs Monte Carlo Methods:

e.g. Random Walk



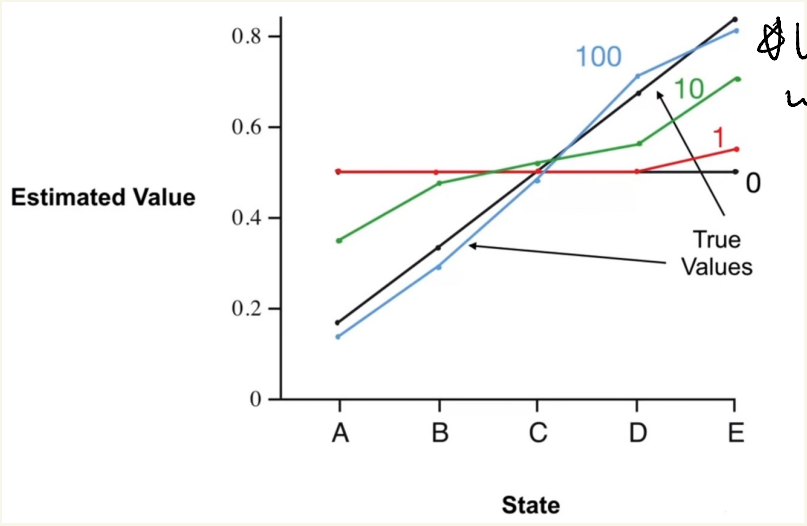
- Value of each state is prob. of terminating on the right if you are at that state.
- Reward is +1 for terminating on right, else 0.
- Always start at C.
- Random chance of moving left/right.

After a Few Episodes!



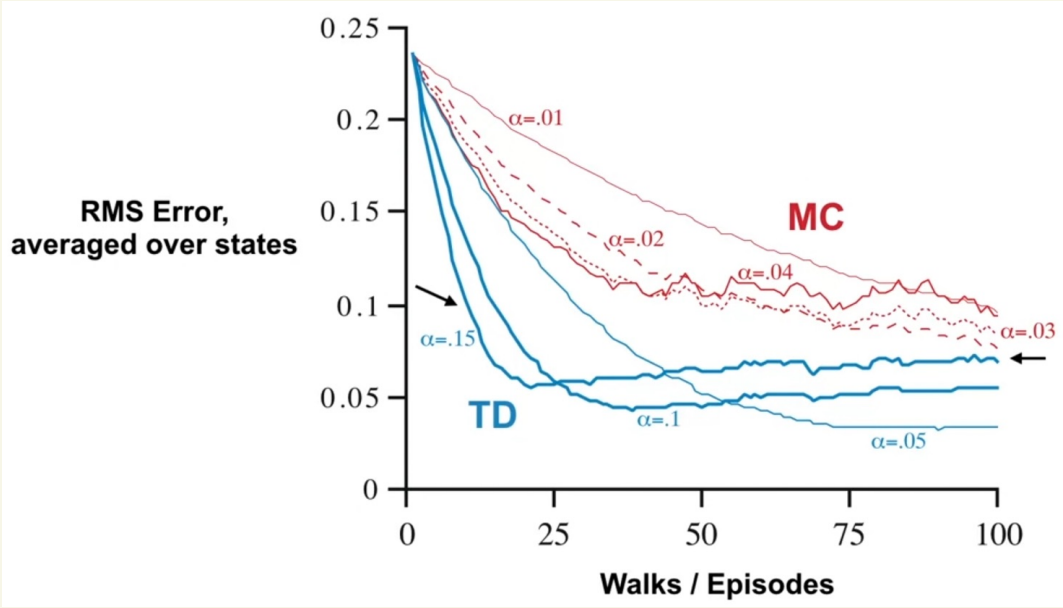
* TD is doing better

TD Performance Over # Episodes!



* World approach closer w/ smaller α learning rate.

Results:



∴ TD converges to a lower final error for this problem.

Note:

TD can also be used in continuing tasks because they update incrementally via bootstrapping, unlike Monte Carlo methods.