TEMPORAL DIFFERENCE LEARNING

Zack Khan and Kevin Chen

University of Maryland

CMSC389F: Reinforcement Learning, Spring 2018

LEARNING GOALS

- Further understand benefit of Model Free algorithms over DP algorithms
- Understand limitations of Monte Carlo approach
- Understand TD Learning

APRIL/MAY

- Q-Learning
- Programming-oriented lectures
- Final Project

No Problem Set this week!

Next week’s problem set will include both TD Learning and Q-Learning

WHY MODEL-FREE?

- Dynamic programming (value/policy iteration) requires **complete knowledge** about the environment (MDP)
- In the real-world, we don't have complete knowledge about the environment
- Less dependant on size of the state space (not doing an entire sweep), and focused on samples

MODEL-FREE RL

Previously: Estimate value function without a model of the environment using Monte Carlo

Now: Estimate value function without a model of the environment using TD Learning

LIMITATIONS OF MONTE-CARLO

- Monte Carlo only from COMPLETE episodes (episodes with terminal state at end)
- If you are driving a car, and you think you're going to crash, but you barely miss it, Monte Carlo is not going to see that! It's only going to see that you didn't crash in the end, not that you almost crashed on your way there. It is **outcome driven.**

TD LEARNING

- TD Learning = Temporal Difference Learning

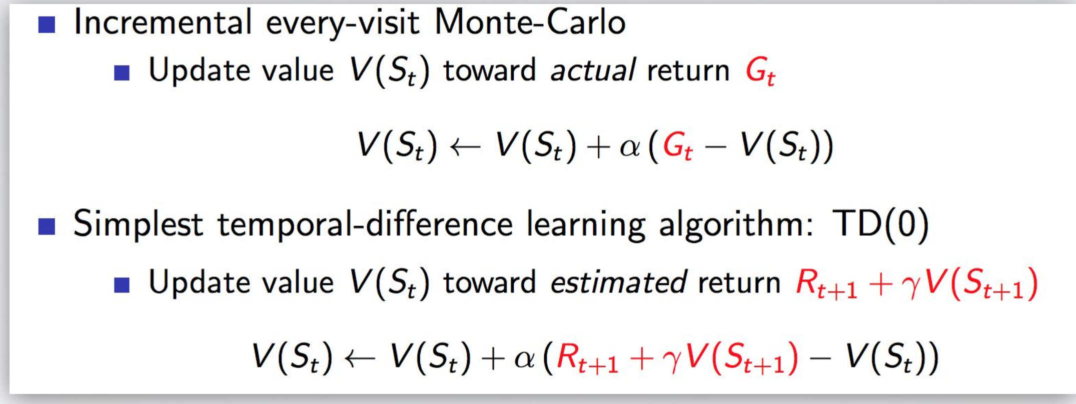
- Learn from incomplete episodes by **bootstrapping:** substituting the remainder of our return (aka cumulative reward) with an estimate.

- Bootstrapping is updating our guess of the value function using a guess of the subsequent value (lookahead).

- Intuition: I can look at future values (lookahead) and update my previous values based on that.

TD LEARNING

In Monte Carlo, we used the actual return (actual sum of rewards we received afterwards)



- Incremental every-visit Monte-Carlo
  - Update value $V(S_t)$ toward *actual* return G_t

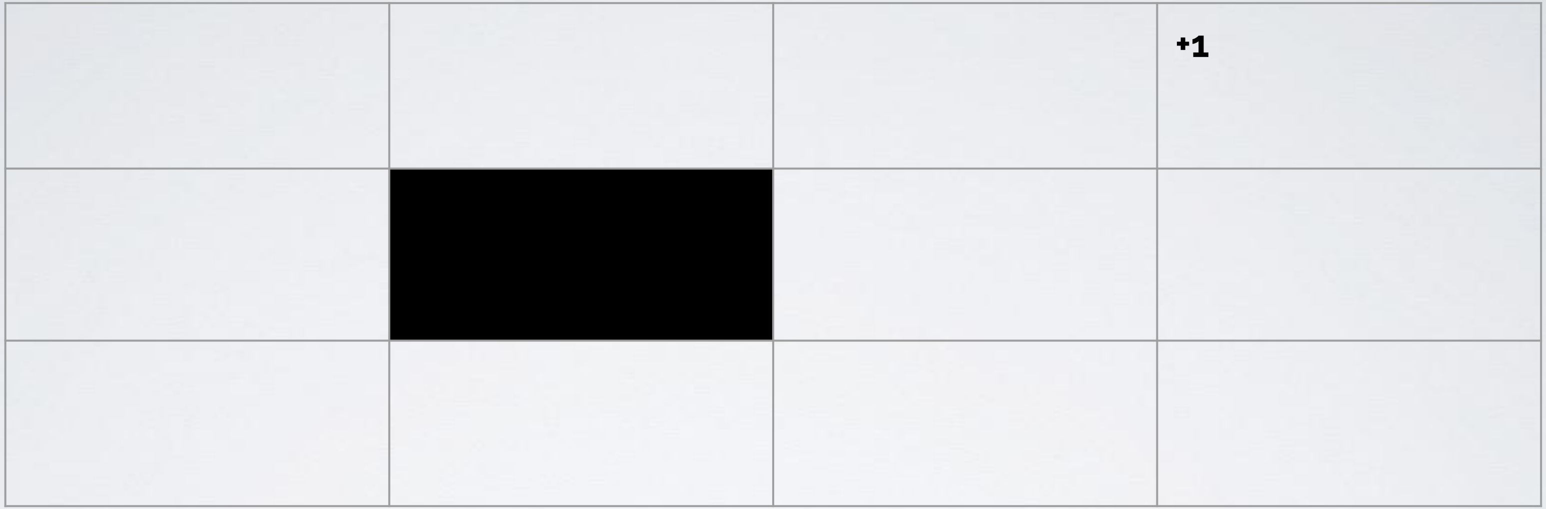
$$V(S_t) \leftarrow V(S_t) + \alpha\left(G_t - V(S_t)\right)$$

- Simplest temporal-difference learning algorithm: TD(0)
  - Update value $V(S_t)$ toward *estimated* return $R_{t+1} + \gamma V(S_{t+1})$

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

For TD(0), we replace the actual return, Gt, with our estimate for the next state.

TEMPORAL DIFFERENCE EXAMPLE



We’ll be updating values of **prior states**

Episode: 1 -> 5 -> 8 -> 9 -> 10 -> 11
Reward: 0

TEMPORAL DIFFERENCE EXAMPLE

			+1
	■		

Learn rate: Determines how much we should learn from this new episode *(high for states we have rarely seen, and low for states we've seen a lot)*

Episode: 1 -> 5 -> 8 -> 9 -> 10 -> 11
Reward: 0

V(S) = V(S) + Learn Rate * (Reward + Gamma * V(S') - V(S))
Learn Rate = 1/(n+1)

TEMPORAL DIFFERENCE EXAMPLE

0	**0**	**0**	**+1**
0	■		
0			

Learn rate: Determines how much we should learn from this new episode *(high for states we have rarely seen, and low for states we've seen a lot)*

Episode: 1 -> 5 -> 8 -> 9 -> 10 -> 11
Reward: 0

V(S) = V(S) + Learn Rate * (Reward + Gamma * V(S') - V(S))
Learn Rate = 1/(n+1)

TEMPORAL DIFFERENCE EXAMPLE

0	**0**	**1/2**	**+1**
0	■		
0			

V(S) = 0 + ½ (0 + 1- 0) = 1/2
Learn Rate = 1/(n+1) = 1(1+1) = ½

Learn rate: Determines how much we should learn from this new episode *(high for states we have rarely seen, and low for states we’ve seen a lot)*

Episode: 1 -> 5 -> 8 -> 9 -> 10 -> 11
Reward: 0

V(S) = V(S) + Learn Rate * (Reward + Gamma * V(S’) - V(S))
Learn Rate = 1/(n+1)

TEMPORAL DIFFERENCE EXAMPLE

0	**1/6**	**1/2**	**+1**
0	■		
0			

V(S) = 0 + 1/3 (0 + 1/2- 0) = 1/6
Learn Rate = 1/(n+1) = 1(1+2) = ⅓

Learn rate: Determines how much we should learn from this new episode *(high for states we have rarely seen, and low for states we've seen a lot)*

Episode: 1 -> 5 -> 8 -> 9 -> 10 -> 11
Reward: 0
Episode: 1 -> 5 -> 8 -> 9 -> 10

V(S) = V(S) + Learn Rate * (Reward + Gamma * V(S') - V(S))
Learn Rate = 1/(n+1)

TD VS MONTE CARLO

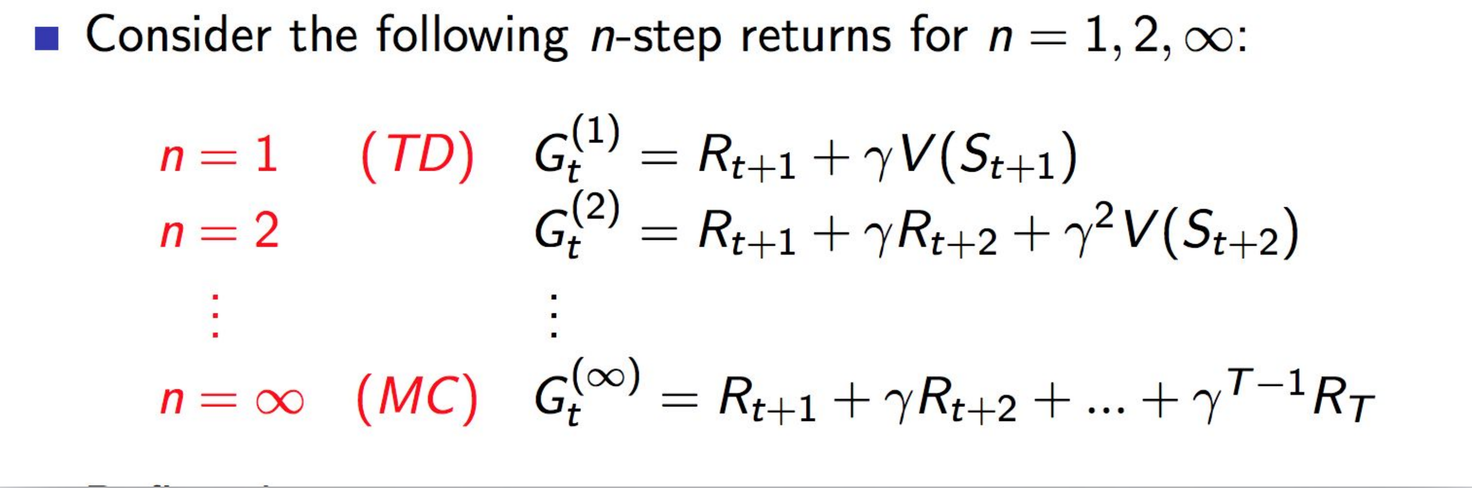
TEMPORAL DIFFERENCE

- TD can learn before knowing the final outcome of an episode
- TD can learn online after every step
- TD can learn without the final outcome
- TD can learn from incomplete sequences

MONTE CARLO

- MC must wait until end of episode before return is known
- MC can only learn from complete sequences TD works in continuing (non-terminating) environments
- MC only works for episodic (terminating) environments

MORE THAN ONE STEP LOOKAHEAD?



- Consider the following n-step returns for $n = 1, 2, \infty$:

$$
\begin{array}{lll}
n = 1 & (TD) & G_t^{(1)} = R_{t+1} + \gamma V(S_{t+1}) \\
n = 2 & & G_t^{(2)} = R_{t+1} + \gamma R_{t+2} + \gamma^2 V(S_{t+2}) \\
\vdots & & \vdots \\
n = \infty & (MC) & G_t^{(\infty)} = R_{t+1} + \gamma R_{t+2} + ... + \gamma^{T-1} R_T
\end{array}
$$

NOTE: TD Learning with infinite lookahead = Monte Carlo

MIDTERM DUE BY MIDNIGHT **TONIGHT**