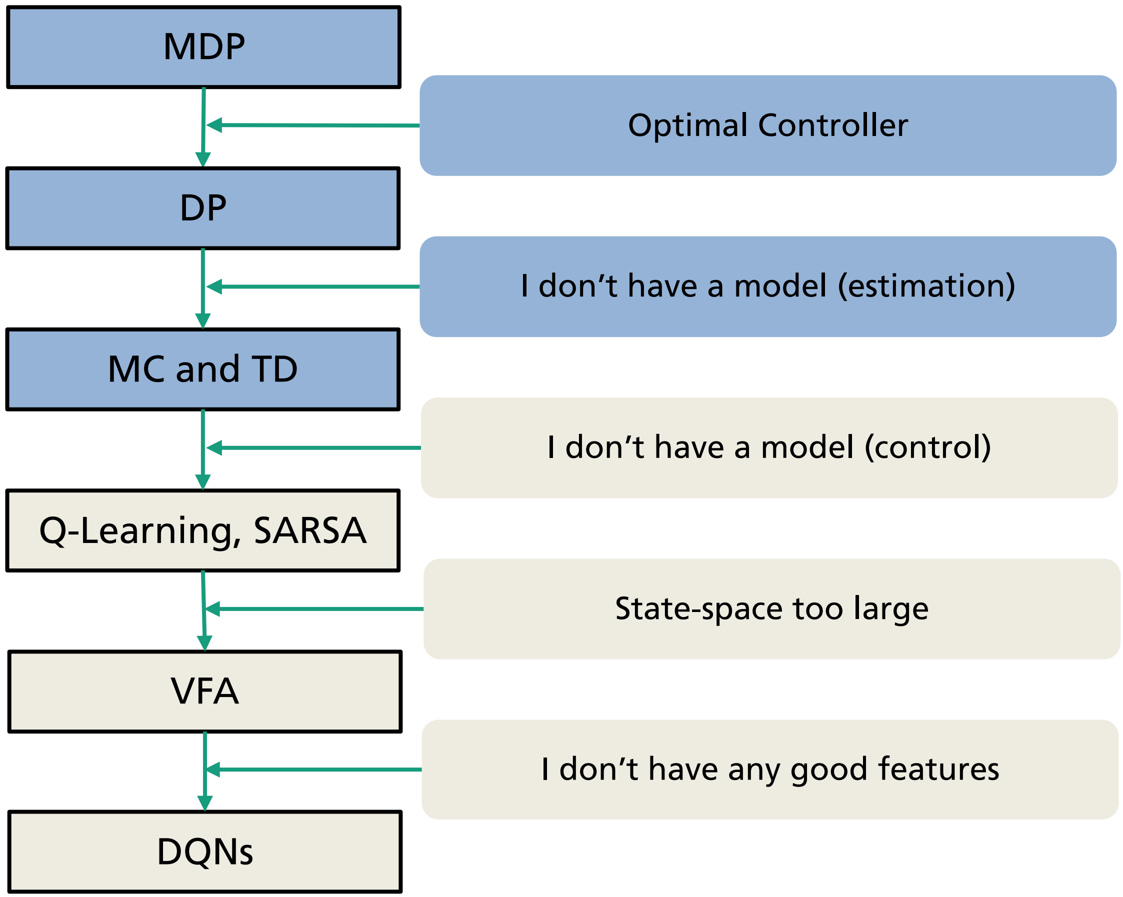
Model-free Prediction: TD-Learning

Christopher Mutschler

Overview


MDP
Optimal Controller
DP
I don’t have a model (estimation)
MC and TD
I don’t have a model (control)
Q-Learning, SARSA
State-space too large
VFA
I don’t have any good features
DQNs

Monte Carlo and TD Methods

Assumptions:

- We know that the model of the world can be described by an MDP:

$$(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$$

- We know the (discrete) state and action spaces, i.e., $\mathcal{S}$ and $\mathcal{A}$.
- We can interact with the world (with some policy π).
- We receive experience samples from the environment in the form

$$(S_t, A_t, R_t, S_{t+1}) = (s, a, r, s').$$

Monte Carlo and TD Methods

- Temporal-Difference Learning
  - Breaks up episodes and makes use of the intermediate returns
  - Learns directly from experience and interaction with the environment
  - Model-free: no knowledge of MDP
  - Learns from incomplete episodes (bootstrapping)
  - **We update a guess towards a guess**

Monte Carlo and TD Methods

- Temporal-Difference Learning: Idea of TD(0) Policy Evaluation

$$V^{\pi}(s) = \underbrace{r(s,\pi(s))} + \gamma \underbrace{\sum_{s' \in S} \boxed{\mathcal{P}(s'|s,\pi(s))} V^{\pi}(s')}$$

We don't know the transition model

$$\boxed{(s,a,r,s')}$$

But we have real transitions available

$$\boxed{V^{\pi}(s) = r + \gamma V^{\pi}(s')}$$

Let's assume that the reality is the transition we observed

$$\boxed{V(s) \leftarrow V(s) + \alpha(r + \gamma V(s') - V(s))}$$

→ and update our old estimate "a bit" in this direction

Monte Carlo and TD Methods

- TD(0) vs. MC Policy Evaluation
  - Goal: learn value function v_π online from experience when we follow policy π

- Simplest TD learning algorithm: TD(0)
- Update value **towards estimation $\hat{G}$**:

$$V(s) \leftarrow V(s) + \alpha(\hat{\boldsymbol{G}} - V(s))$$
$$\hat{\boldsymbol{G}} = \boldsymbol{r} + \boldsymbol{\gamma V(s')} \text{ (estimated return)}$$

- $\hat{G}$ is called the TD target
- $\hat{G} - V(s)$ is called the TD error.

- Update $V(s)$ incrementally after each episode.
- For each state s with **actual return $\boldsymbol{G}$**:

$$N(s) \leftarrow N(s) + 1 \text{ (just increment visit counter)}$$
$$V(s) \leftarrow V(s) + \frac{1}{N(s)}\left(\boldsymbol{G} - V(s)\right) \text{ (update a bit → reduce error)}$$

- In non-stationary problems, it can be useful to track a running mean, i.e., forget old episodes:

$$V(s) \leftarrow V(s) + \alpha\left(\boldsymbol{G} - V(s)\right).$$

$$\underbrace{\hphantom{xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx}}$$

$$NewEstimate \leftarrow OldEstimate + StepSize\,[Target - OldEstimate]$$

Monte Carlo and TD Methods

- TD(0) vs. MC Policy Evaluation

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'
    $V(S) \leftarrow V(S) + \alpha\big[R + \gamma V(S') - V(S)\big]$
    $S \leftarrow S'$
  until S is terminal

First-visit MC prediction, for estimating $V \approx v_\pi$

Input: a policy π to be evaluated
Initialize:
  $V(s) \in \mathbb{R}$, arbitrarily, for all $s \in \mathcal{S}$
  $Returns(s) \leftarrow$ an empty list, for all $s \in \mathcal{S}$

Loop forever (for each episode):
  Generate an episode following π: $S_0, A_0, R_1, S_1, A_1, R_2, \ldots, S_{T-1}, A_{T-1}, R_T$
  $G \leftarrow 0$
  Loop for each step of episode, $t = T-1, T-2, \ldots, 0$:
    $G \leftarrow \gamma G + R_{t+1}$
    Unless S_t appears in $S_0, S_1, \ldots, S_{t-1}$:
      Append G to $Returns(S_t)$
      $V(S_t) \leftarrow \text{average}(Returns(S_t))$

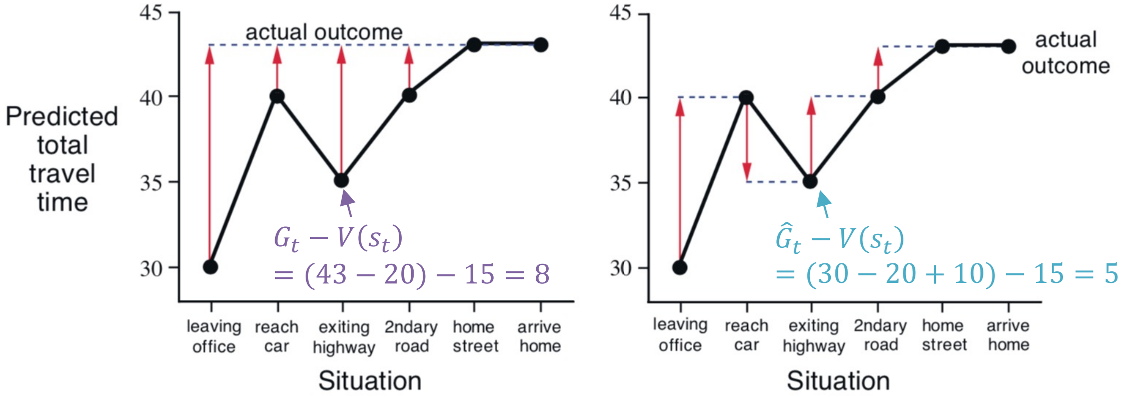
Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction. MIT press.

Monte Carlo and TD Methods

- Which one should I use? Does it make any difference?
- Example: Driving Home from work

State	*Elapsed Time [min]*	*Predicted Time to Go [min]*	*Predicted Total Time [min]*
leaving office, friday at 6	0	30	30
reach car, raining	5	35	40
exiting highway	20	15	35
2ndary road, behind truck	30	10	40
entering home street	40	3	43
arrive home	43	0	43

MC ($\alpha = 1$)

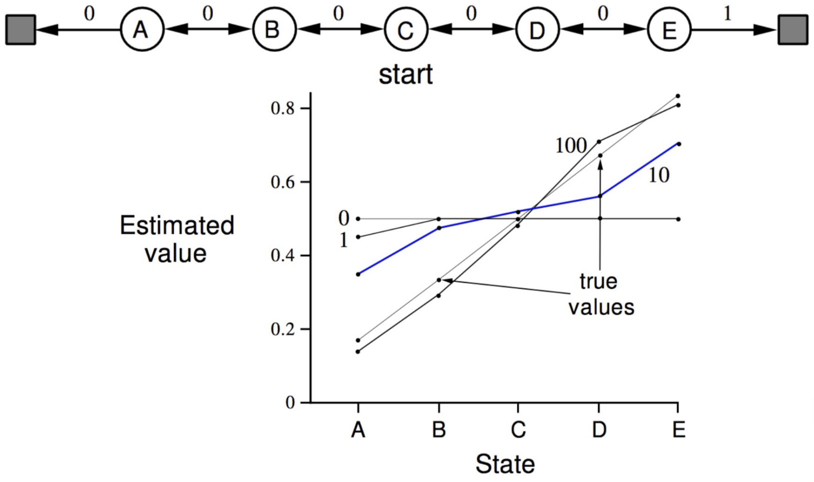


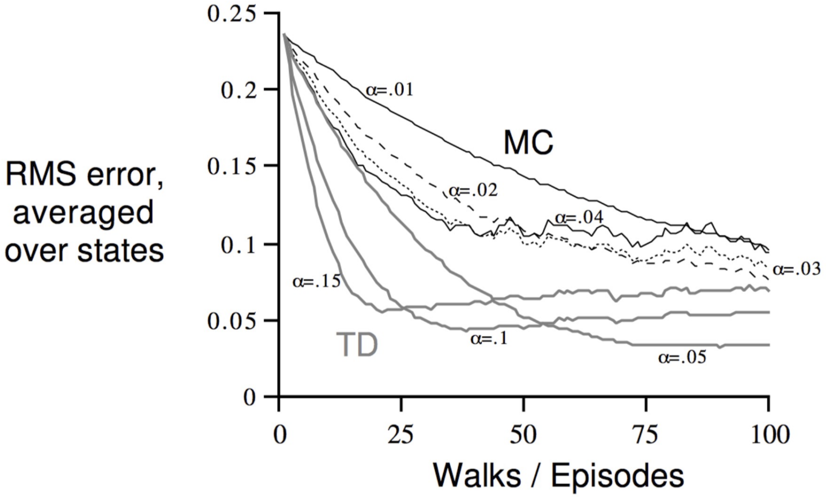
TD ($\alpha = 1$)

Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction. MIT press.

Monte Carlo and TD Methods

- Which one should I use? Does it make any difference?
- Example: Random Walk





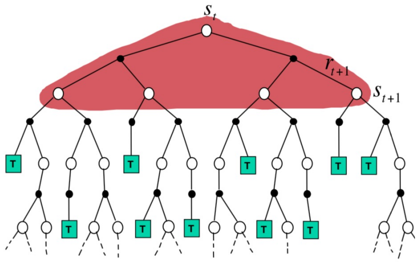
Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction. MIT press.

Monte Carlo and TD Methods

- Which one should I use? Does it make any difference?
  - TD can learn before (or even without) knowing the final outcome
    - after each step
    - incomplete sequences
    - continuing problems, very delayed or no return
  - MC only works for episodic problems (i.e., that terminate)
    - must wait until end of the episode

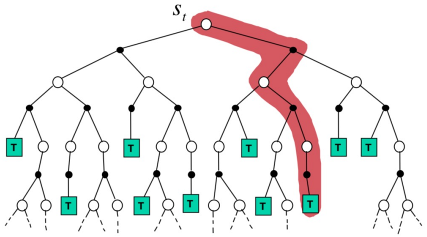
<u>DP Backup</u>

$V(S_t) \leftarrow \mathbb{E}_\pi\left[R_{t+1} + \gamma V(S_{t+1})\right]$



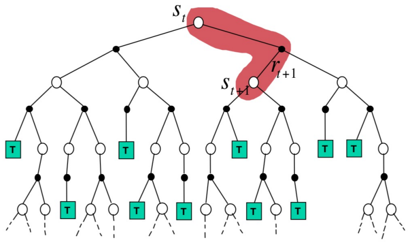
<u>MC Backup</u>

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



<u>TD Backup</u>

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



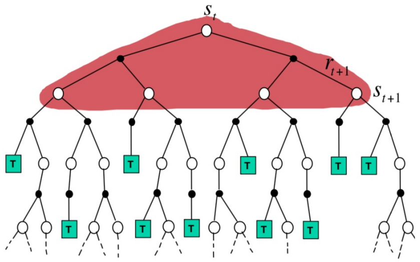
Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction. MIT press.

Monte Carlo and TD Methods

- Which one should I use? Does it make any difference?
  - Bias/Variance Trade-Off
  - MC has high variance, but zero bias
    - Good convergence (even with FA)
    - insensitive to initialization (no bootstrapping), simple to understand
  - TD has low variance, but some bias
    - TD(0) converges to $\pi_v(s)$ (be careful with FA: bias is a risk)
    - sensitive to initialization (because of the bootstrapping)
    - Usually more efficient in practice

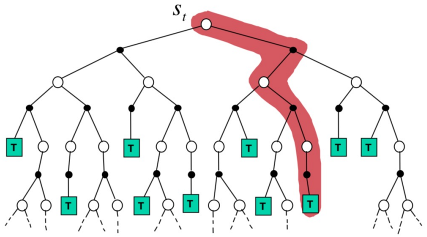
DP Backup

$V(S_t) \leftarrow \mathbb{E}_\pi[R_{t+1} + \gamma V(S_{t+1})]$



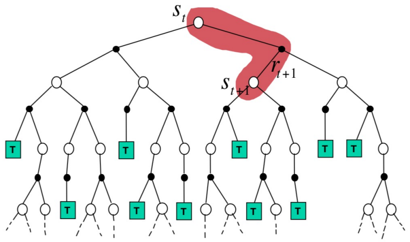
MC Backup

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



TD Backup

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

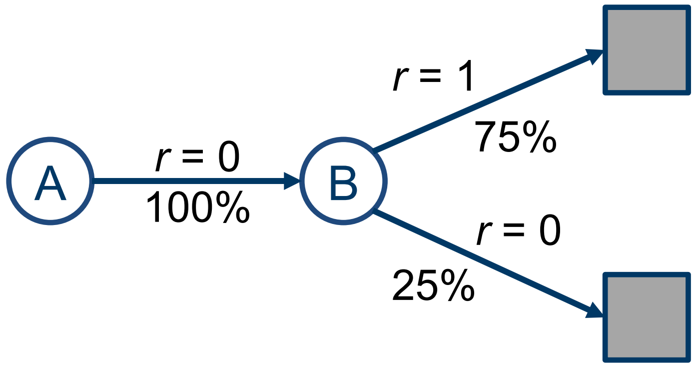


Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction. MIT press.

Monte Carlo and TD Methods

- Which one should I use? Does it make any difference?
- Example: You are the predictor!
  - Two states A, B; no discounting; 8 episodes of experience
  - keep iterating on experience (MC and TD until both of them converge):

```
A, 0, B, 0
B, 1
B, 1
B, 1
B, 1
B, 1
B, 1
B, 0
```



Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction. MIT press.

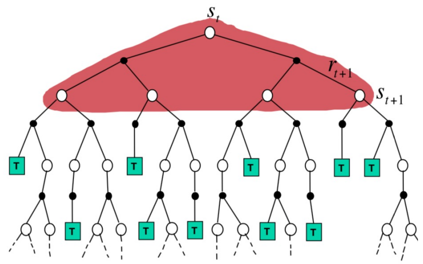
  - What is $V(S = A)$ and $V(S = B)$?
    - MC: $V(A) = 0$ $V(B) = 0.75$
    - TD: $V(A) = 0.75$ $V(B) = 0.75$

Monte Carlo and TD Methods

- Which one should I use? Does it make any difference?
  - TD exploits Markov property and is more efficient in Markov environments
  - MC is more efficient in non-Markov environments
  - TD usually converges faster than MC

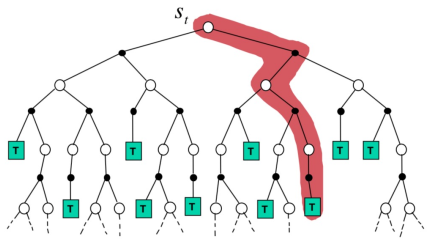
<u>DP Backup</u>

$V(S_t) \leftarrow \mathbb{E}_\pi\left[R_{t+1} + \gamma V(S_{t+1})\right]$



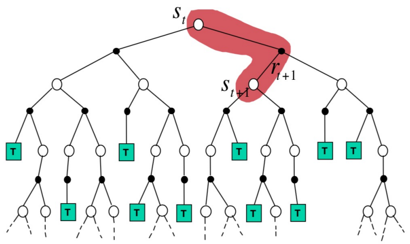
<u>MC Backup</u>

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



<u>TD Backup</u>

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



Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction. MIT press.

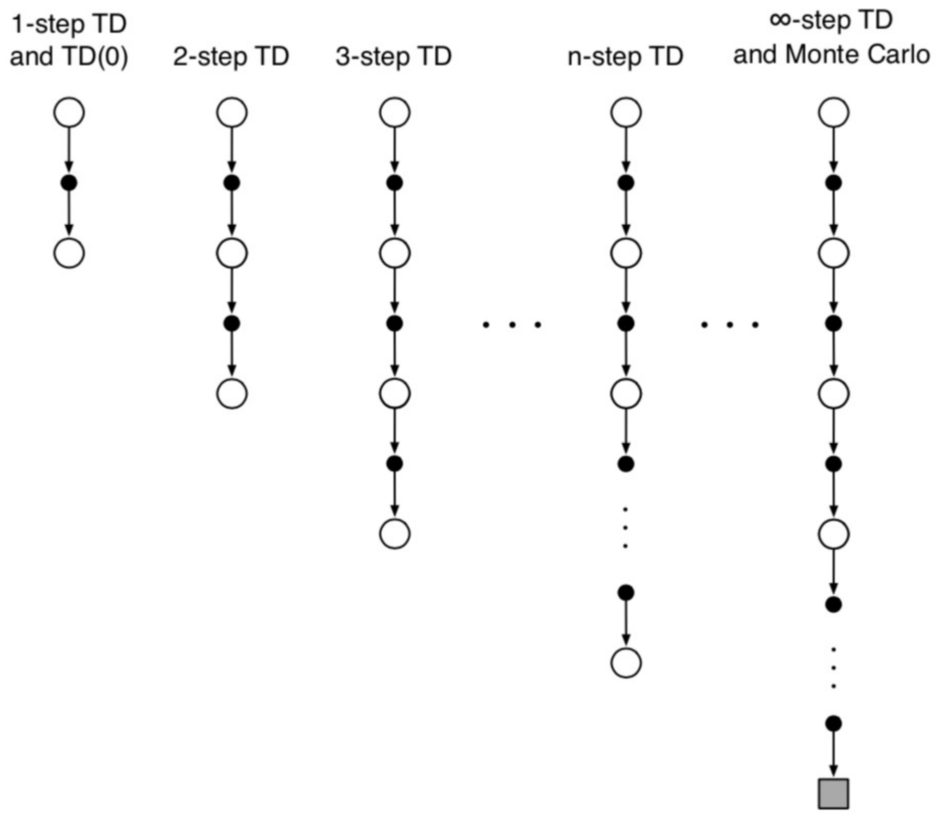
Monte Carlo and TD Methods

Hands-On:

https://cs.stanford.edu/people/karpathy/reinforcejs/gridworld_td.html

Monte Carlo and TD Methods

- Intermediate methods between MC and TD(0) exist
- They are based on n-step returns



Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction. MIT press.

Monte Carlo and TD Methods

- Intermediate methods between MC and TD(0) exist
- They are based on n-step returns

n-step TD for estimating $V \approx v_\pi$

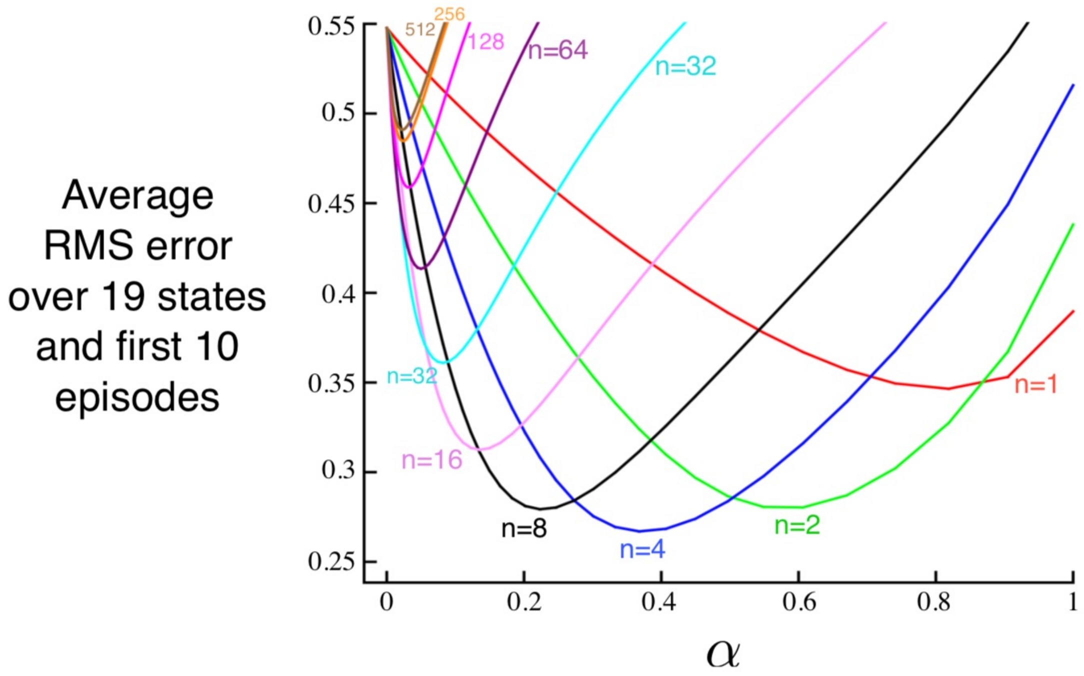
Input: a policy π
Algorithm parameters: step size $\alpha \in (0,1]$, a positive integer n
Initialize $V(s)$ arbitrarily, for all $s \in \mathcal{S}$
All store and access operations (for S_t and R_t) can take their index mod $n+1$

Loop for each episode:
    Initialize and store $S_0 \neq$ terminal
    $T \leftarrow \infty$
    Loop for $t = 0, 1, 2, \dots$:
    | If $t < T$, then:
    | Take an action according to $\pi(\cdot|S_t)$
    | Observe and store the next reward as R_{t+1} and the next state as S_{t+1}
    | If S_{t+1} is terminal, then $T \leftarrow t+1$
    | $\tau \leftarrow t - n + 1$ (τ is the time whose state's estimate is being updated)
    | If $\tau \geq 0$:
    | $G \leftarrow \sum_{i=\tau+1}^{\min(\tau+n,T)} \gamma^{i-\tau-1} R_i$
    | If $\tau + n < T$, then: $G \leftarrow G + \gamma^n V(S_{\tau+n})$ $(G_{\tau:\tau+n})$
    | $V(S_\tau) \leftarrow V(S_\tau) + \alpha \left[G - V(S_\tau)\right]$
    Until $\tau = T - 1$

Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction. MIT press.

Monte Carlo and TD Methods

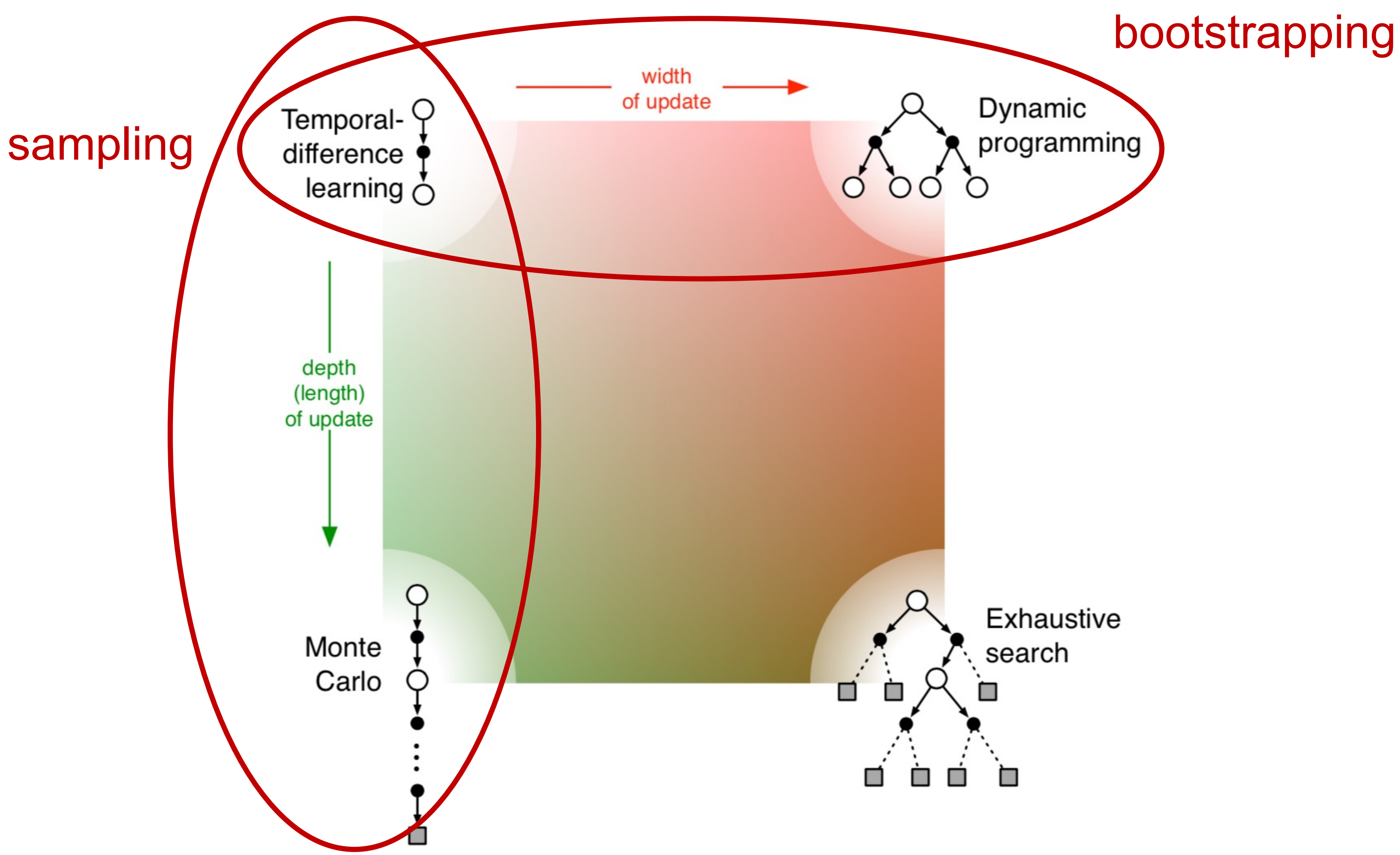
- Intermediate methods between MC and TD(0) exist
- They are based on n-step returns
- Unfortunately, their prediction accuracy is sensitive to the algorithm hyperparameters
- Example: Random Walk



Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction. MIT press.

Monte Carlo and TD Methods

- A Unified View of Prediction Algorithms



Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction. MIT press.

TD – Remarks on Convergence Properties

- There is a lot of work that studied the convergence of TD:
- Convergence and optimality of (linear) TD methods under batch training (no online learning):
  - Richard S. Sutton: Learning to predict by the Methods of Temporal Differences. Machine Learning 3:9-44. 1988.
- Build on [Sutton1988] and proofs convergence of TD(0) and extends Watkin's Q-learning theorem (next video):
  - Peter Dayan: The Convergence of TD(λ) for General λ. Machine Learning 8:341-362. 1992.
- Further studies in the context of Q-Learning and SARSA (next video):
  - Tommi Jaakkola, Michael Jordan, Stainder Singh: On the Convergence of Stochastic Iterative Dynamic Programming Algorithms. Technical report. 1994.
  - Francisco Melo: Convergence of Q-Learning: A Simple Proof. Technical report.
    (it has only 4 pages – so feel free to have a look ☺)
  - Satinder Singh, Tommi Jakkola, Michael Littman, Csaba Szepesvari: Convergence Results for Single-Step On-Policy Reinforcement-Learning Algorithms. Machine Learning 39:287-308. 2000.