1

Model-Free Prediction

CMPS 4660/6660: Reinforcement Learning

Acknowledgement: slides adapted from David Silver's RL course

Agenda

- Monte Carlo Method
- TD(0)
- n-step TD
- TD(λ)

} Temporal-difference (TD) learning

Model-free reinforcement learning

- Planning by dynamic programming
  - Solve a *known* MDP
- Model-free prediction
  - Estimate the value function of an *unknown* MDP
- Model-free control
  - Optimize the value function of an *unknown* MDP

Monte-Carlo Reinforcement Learning

- MC methods learn directly from episodes of experience
  - Sample sequences of states, action, rewards from actual or simulated interaction with an environment
  - Model-free: no knowledge of MDP transitions/rewards
- MC uses the simplest possible idea: value = mean return
- MC learns from complete episodes
  - no bootstrapping
  - only applies to episodic MDPs that always terminate

Monte-Carlo Policy Evaluation

- Goal: learn v_π from episodes of experience under a stationary policy π

$$S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T \sim \pi$$

- Recall that the *return* is the total discounted reward:

$$G_t \doteq R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{T-1} R_T$$

- Recall that the value function is the expected return:

$$v_\pi(s) \doteq \mathbb{E}_\pi(G_t | S_t = s)$$

- Monte-Carlo policy evaluation uses empirical mean return
  - instead of expected return (which is unknown)

First-Visit Monte-Carlo Policy Evaluation

- To evaluate state s
- The first time-step t that state s is visited in an episode
- Increment counter $N(s) \leftarrow N(s) + 1$
- Increment total return $S(s) \rightarrow S(s) + G_t$
- Value is estimated by mean return $V(s) \leftarrow S(s)/N(s)$
- By law of large numbers, $V(s) \rightarrow v_\pi(s)$ as $N(s) \rightarrow \infty$
  - $\mathrm{E}\big(V(s)\big) = \; v_\pi(s)$: $V(s)$ is an unbiased estimate of $v_\pi(s)$
  - $\sqrt{\mathrm{Var}(V(s))} = \sigma/\sqrt{N(s)}$: rate of convergence is $1/\sqrt{N(s)}$

First-Visit Monte-Carlo Policy Evaluation

Input: a policy π to be evaluated

Initialize:

$V(s) \in \mathbb{R}$, arbitrarily, for $s \in \mathcal{S}$

$Return(s) \leftarrow$ an empty list for for $s \in \mathcal{S}$

Loop forever (for each episode):

Generate an episode following π: $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$

$G \leftarrow 0$

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 $Return(S_t)$

$V(S_t) \leftarrow$ average$\big(Return(S_t)\big)$

Every-Visit Monte-Carlo Policy Evaluation

- To evaluate state s
- Every time-step t that state s is visited in an episode
- Increment counter $N(s) \leftarrow N(s) + 1$
- Increment total return $S(s) \rightarrow S(s) + G_t$
- Again $V(s) \rightarrow v_\pi(s)$ as $N(s) \rightarrow \infty$
  - See Singh and Sutton, "Reinforcement learning with replacing eligibility traces", 1996

Incremental Mean

- Let $(X_n)_{n\geq 0}$ be an $i.i.d.$ sequence of random variables with mean $\mu = E[X_0]$
- Let θ_n be the empirical mean of $X_1, X_2, \dots, X_n$
- θ_n can be computed incrementally

$$\theta_{n+1} = \frac{1}{n+1}\sum_{i=1}^{n+1} X_i$$

$$= \frac{1}{n+1}\left(X_{n+1} + \sum_{i=1}^{n} X_i\right)$$

$$= \frac{1}{n+1}(X_{n+1} + n\theta_n)$$

$$= \theta_n + \frac{1}{n+1}(X_{n+1} - \theta_n)$$

Incremental Monte-Carlo Updates

- Update $V(s)$ after each episode $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$
- For each state S_t with return G_t:

$$N(S_t) \leftarrow N(S_t) + 1$$

$$V(S_t) \leftarrow V(S_t) + \frac{1}{N(S_t)}\left(G_t - V(S_t)\right)$$

- Constant-α MC: $V(S_t) = V(S_t) + \alpha\left(G_t - V(S_t)\right)$
  - Useful in non-stationary problems to track a running mean, i.e. forget old episodes
  - A special case of Widrow-Hoff learning rule (1960)
- MC with a general stepsize: $V(S_t) = V(S_t) + \alpha\left(N(S_t)\right)\left(G_t - V(S_t)\right)$

Estimation of Mean

- Let $(X_n)_{n\geq 0}$ be an $i.i.d.$ sequence of random variables with mean $\mu = E[X_0]$ and a bounded variance
- Consider the estimator: $\theta_{n+1} = \theta_n + \alpha_n(X_{n+1} - \theta_n)$
- Theorem: if $\sum_{n\geq 0} \alpha_n = \infty$ and $\sum_{n\geq 0} \alpha_n^2 < \infty$, then $\theta_n \to \mu$ almost surely, that is, $\Pr\left(\lim_{n\to\infty} \theta_n = \mu\right) = 1.$
  - A common example: $\alpha_n = \frac{1}{n^a}$ with $\frac{1}{2} < a \leq 1$
- For constant stepsize α that is small enough, $\limsup_{n\to\infty} \Pr(\|\theta_n - \mu\| > \epsilon) \leq b(\epsilon) \cdot \alpha$, with $b(\epsilon) < \infty$.

Estimation of Mean as Stochastic Approximation

$$\theta_{n+1} = \theta_n + \alpha_n(X_{n+1} - \theta_n)$$

$$= \theta_n + \alpha_n[\mu + (X_{n+1} - \mu) - \theta_n]$$

$$= \theta_n + \alpha_n\,[\mu + \omega_n - \theta_n]$$

$\omega_n \doteq X_{n+1} - \mu$: *i.i.d.* & zero mean

$$= \theta_n + \alpha_n[\mu - \theta_n + \omega_n]$$

$$= \theta_n + \alpha_n[h(\theta_n) + \omega_n]$$

$h(\theta_n) \doteq \mu - \theta_n$

Want to find θ^* such that $h(\theta^*) = 0$ from noisy observations $h(\theta_n) + \omega_n, n \geq 0$

Stochastic Approximation

- Stochastic Approximation Methods: a family of iterative stochastic optimization algorithms that attempt to find zeroes or extrema of functions which cannot be computed directly, but only estimated via noisy observations.

- The first and prototypical algorithms of this kind are: *Robbins-Monro* (1951) and *Kiefer-Wolfowitz* (1952) algorithms

Robbins-Monro Stochastic Approximation

- We have a function $h(\theta)$ and want to find θ^* such that $h(\theta^*) = 0$
- But only have noisy observations $Y_n = h(\theta_n) + \omega_n$
- SA algorithm:

$$\begin{aligned}\theta_{n+1} &= \theta_n + \alpha_n Y_n \\ &= \theta_n + \alpha_n[h(\theta_n) + \omega_n], \qquad n \geq 0\end{aligned}$$

- The same framework applies to MC, TD, Q-learning, and other RL algorithms
  - MC: $h(\theta) \doteq \mu - \theta$
  - TD(0): $h(\theta) \doteq T^\pi(\theta) - \theta$

Function Minimization via Stochastic Approximation

- Suppose we wish to minimize a (convex) function $f(\theta)$. Define $h(\theta) = -\nabla \mathrm{f}(\theta) = -\frac{\partial f}{\partial \theta}$, we need to solve $h(\theta) = 0$.
- The basic iteration is

$$\theta_{n+1} = \theta_n + \alpha_n[-\nabla \mathrm{f}(\theta) + \omega_n], \qquad n \geq 0$$

- This is a "noisy" version of gradient descent algorithm.

Stochastic Approximation and ODE

- A common approach to prove the convergence of SA algorithms is to consider the ordinary differential equation (ODE):

$$\frac{d}{dt}\theta(t) = h\big(\theta(t)\big) \text{ or } \dot{\theta} = h(\theta)$$

- Under suitable conditions on $h(\theta), \{\omega_n\}$ and diminishing $\{\alpha_n\}, \{\theta_n\}$ asymptotically "track" a trajectory $\{\theta(t)\}$ of the ODE and converge to a stationary point θ^*: $h(\theta^*) = 0$ of the ODE

- References:
  - https://webee.technion.ac.il/shimkin/LCS11/ch5_SA.pdf
  - H. Kushner and G. Yin, Stochastic Approximation Algorithms and Applications, Springer, 1997.
  - V. Borkar, Stochastic Approximation: A Dynamic System Viewpoint, Hindustan, 2008

Stochastic Approximation (constant stepsize)

- The Robbins-Monro algorithm:

$$\theta_{n+1} = \theta_n + \alpha_n Y_n = \theta_n + \alpha_n[h(\theta_n) + \omega_n]$$

- For constant stepsize $\alpha_n = \alpha$, $\{\theta_n\}$ is a Markov chain. If it is stable, one can only hope $\{\theta_n\}$ has a stationary distribution that assigns a high probability to a neighborhood of θ.
- What can be expected? For all $\epsilon > 0$,

$$\limsup_{n\to\infty} \Pr(||\theta_n - \theta^*|| > \epsilon) \leq \alpha \cdot \mathrm{b}(\epsilon)\text{ , with } \mathrm{b}(\epsilon) < \infty$$

- constant stepsize is more appropriate for nonstationary environment

Improvements of Monte Carlo Method

- Quasi-Monte Carlo method
  - uses non-*i.i.d.* sequence
  - rate of convergence close to $\frac{1}{n}$
  - may have issues for high dimensional random vectors

- Importance Sampling
  - estimates expected values under one distribution given samples from another
  - reduces variance
  - explained later

Agenda

- Monte Carlo Method
- TD(0)
- n-step TD
- TD(λ)

Temporal-Difference Learning

- TD methods learn directly from episodes of experience
- TD is model-free: no knowledge of MDP transitions / rewards
- TD learns from incomplete episodes, by bootstrapping
- TD updates a guess towards a guess

Expressions of Value Function

- Conditional expectation of return:

$$v_\pi(s) = \mathbb{E}_\pi(G_t | S_t = s)$$

- Bellman Equation:

$$v_\pi(s) = \mathbb{E}_\pi\left(R_{t+1} + \gamma\, v_\pi(S_{t+1}) | S_t = s\right)$$

$$v_\pi(s) = \mathbb{E}_\pi\left(R_{t+1} + \gamma\, R_{t+2} + \gamma^2\, v_\pi(S_{t+2}) | S_t = s\right)$$

$$v_\pi(s) = \mathbb{E}_\pi\left(R_{t+1} + \gamma\, R_{t+2} + \gamma^2\, R_{t+3} + \gamma^3\, v_\pi(S_{t+3}) | S_t = s\right)$$

…

MC and TD

- Goal: learn v_π from episodes of experience under policy π
- Incremental every-visit Monte-Carlo
  - Update value $V(S_t)$ toward *actual* return G_t

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

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

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

  - $R_{t+1} + \gamma V(S_{t+1})$ is called the *TD target*
  - $\delta_t = R_{t+1} + \gamma V(S_{t+1}) - V(S_t)$ is called the *TD error*

Tabular TD(0) for estimating v_π

Input: π (policy to be evaluated), $\alpha \in (0,1]$ (step size)

Initialize $V(s)$ for $s \in \mathcal{S}^+$, arbitrarily except $V(s^*) = 0$

Loop for each episode:

Initizlize S

Loop for each step of epsiode:

Choose $A \sim \pi(\cdot\,|S)$

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

MC vs. TD

- TD can learn *before* knowing the final outcome
  - TD can learn online after every step
  - MC must wait until end of episode before return is known

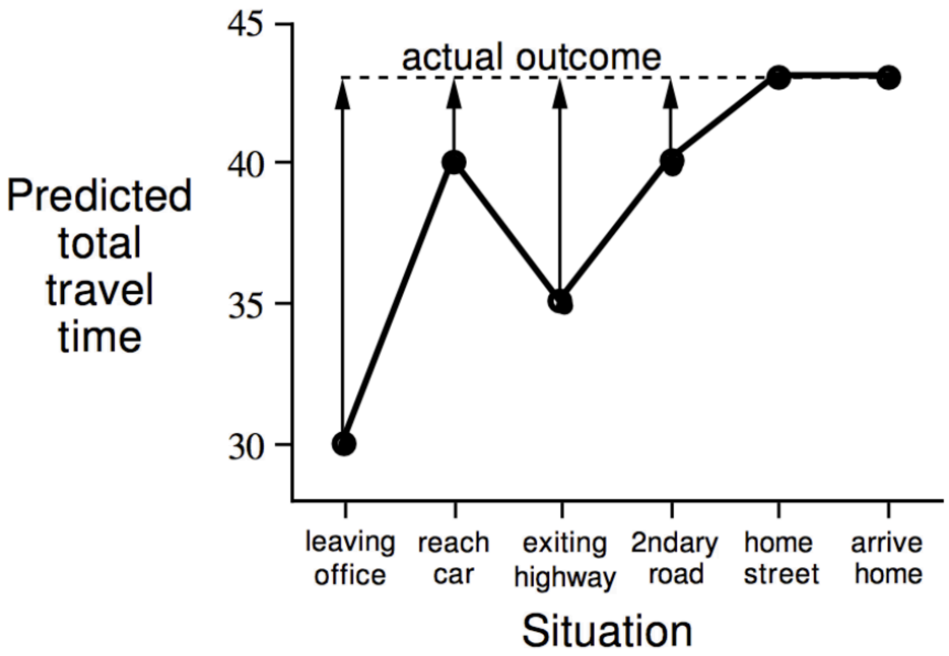
- TD can learn *without* the final outcome
  - TD can learn from incomplete sequences
  - MC can only learn from complete sequences
  - TD works in continuing (non-terminating) environments
  - MC only works for episodic (terminating) environments

Driving Home Example

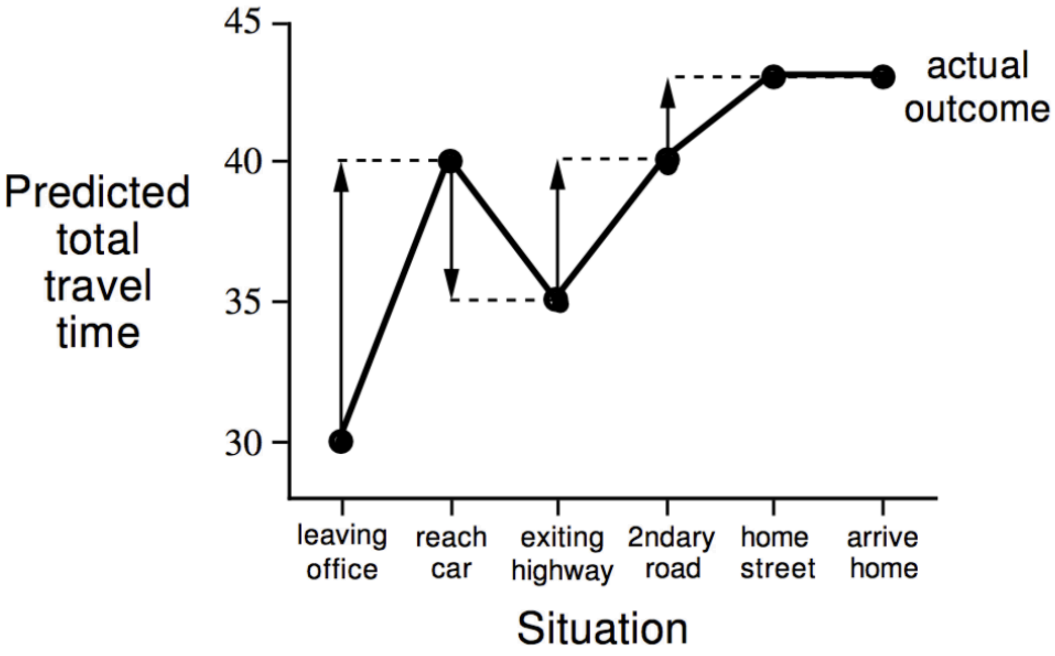
State	Elapsed Time (minutes)	Predicted Time to Go	Predicted Total Time
leaving office, friday at 6	0	30	30
reach car, raining	5	35	40
exiting highway	20	15	35
2ndary road, behind truck	30	10	40
entering home street	40	3	43
arrive home	43	0	43

Driving Home Example: MC vs. TD

Changes recommended by Monte Carlo methods (α=1)



Changes recommended by TD methods (α=1)



TD(0) as Stochastic Approximation

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

$$V_{n+1}(s) = V_n(s) + \alpha_n(s)\,[R_{n+1} + \gamma V_n(S_{n+1}) - V_n(s)] \quad \alpha_n(s) = 0 \text{ if } s \neq S_n$$

$$= V_n(s) + \alpha_n(s)[Z(s, V_n) - V_n(s)] \quad \text{where } Z(s, V_n) \doteq R_{n+1} + \gamma V_n(S_{n+1}) \text{ for } S_n = s$$

$$= V_n(s) + \alpha_n(s)\,[\mathrm{E}_\pi(Z(s, V_n)) - V_n(s) + Z(s, V_n) - \mathrm{E}_\pi(Z(s, V_n))]$$

$$= V_n(s) + \alpha_n(s)[h(s, V_n) + \omega_n(s)]$$

TD(0) as Stochastic Approximation

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

$$V_{n+1}(s) = V_n(s) + \alpha_n(s)[h(s, V_n) + \omega_n(s)]$$

where $h(s, V_n) \doteq \mathrm{E}_\pi(Z(s, V_n)) - V_n(s)$

$$= \mathrm{E}_\pi[R_{n+1} + \gamma V_n(S_{n+1})] - V_n(s)$$

$$= (T^\pi V_n)(s) - V_n(s)$$

a martingale difference sequence

$\omega_n(s) = Z(s, V_n) - \mathrm{E}_\pi(Z(s, V_n))$: zero mean but depend on V_n

TD(0) is an example of asynchronous SA

Theorem: If $\sum_{n\geq 0} \alpha_n(s) = \infty$ and $\sum_{n\geq 0} \alpha_n^2(s) < \infty$ for all s, $\{V_n\}$ converge to the unique solution of $H(V) \doteq T^\pi V - V = 0$

- For the conditions on α to hold, each state should be visited "relatively often"

Bias/Variance Trade-Off

- Return $G_t \doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots + \gamma^{T-t-1} R_T$ is unbiased estimate of $v_\pi(S_t)$
- True TD target $R_{t+1} + \gamma\, v_\pi(S_{t+1})$ is unbiased estimate of $v_\pi(S_t)$
- TD target $R_{t+1} + \gamma\, V(S_{t+1})$ is biased estimate of $v_\pi(S_t)$
- TD target is much lower variance than the return:
  - Return depends on *many* random actions, transitions, rewards
  - TD target depends on *one* random action, transition, reward

MC vs. TD (2)

- MC has high variance, zero bias
  - Good convergence properties
  - (even with function approximation)
  - Not very sensitive to initial value
  - Very simple to understand and use

TD has low variance, some bias

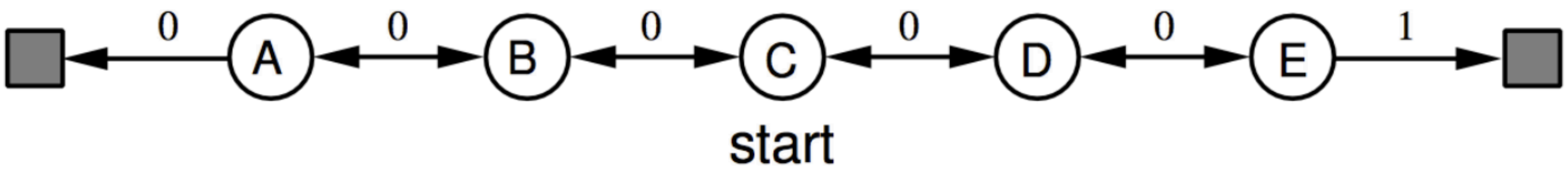
Usually more efficient than MC

TD(0) converges to $v_\pi(s)$

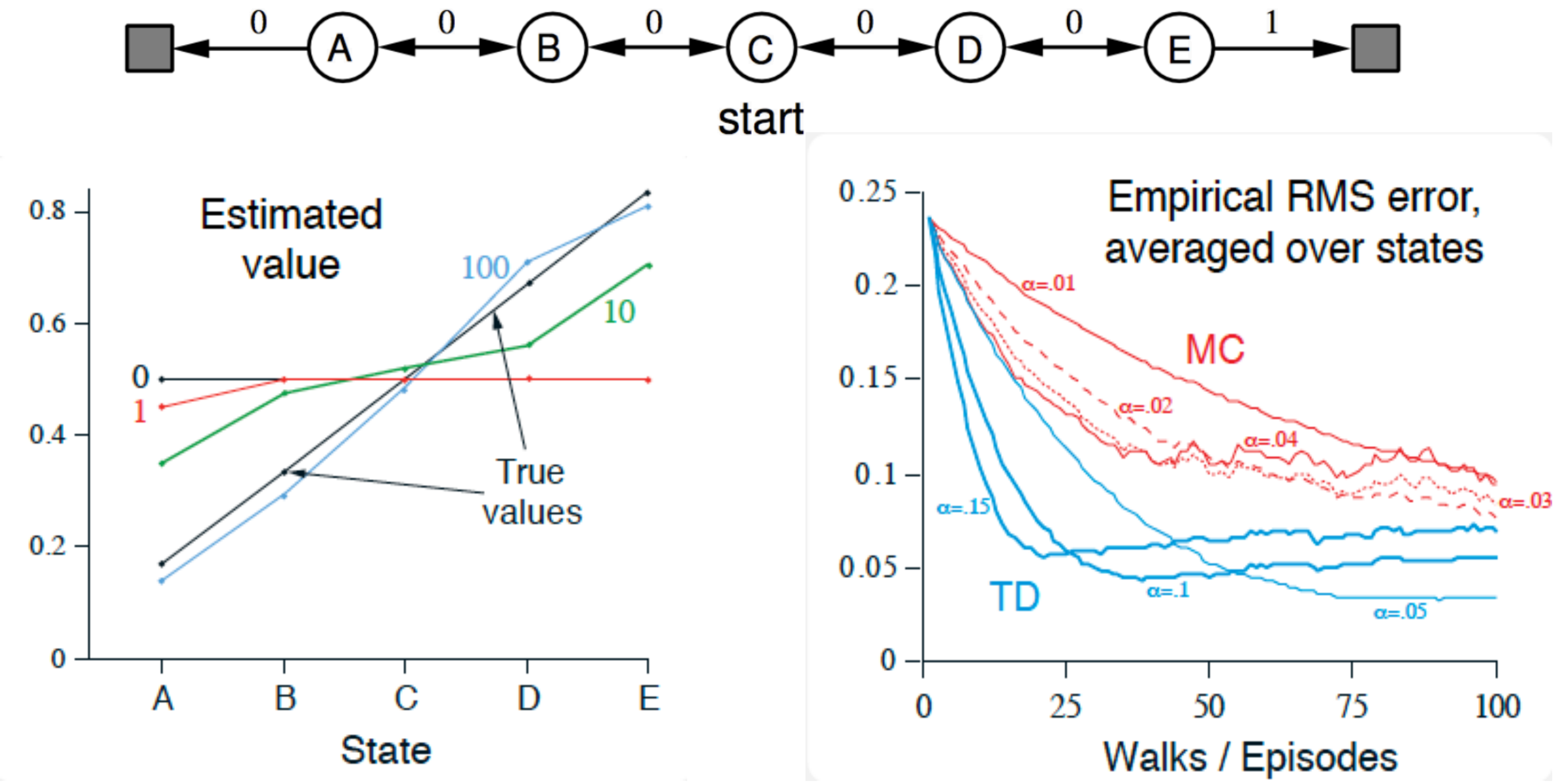
(but not always with function approximation)

More sensitive to initial value

Random Walk Example



Random Walk Example



Values learned after various no. of episodes in TD(0)

Batch MC and TD

- MC and TD converge: $V(s) \rightarrow v_\pi(s)$ as experience $\rightarrow \infty$
- But what about batch solution for finite experience?

$$s_0^1, a_0^1, r_1^1, \dots, S_{T_1}^1$$

$$\vdots$$

$$s_0^K, a_0^K, r_1^K, \dots, S_{T_K}^K$$

- e.g., repeatedly sample episode $k \in \{1, \dots, K\}$
- Apply MC or TD(0) to episode *k*

AB Example

Two states A, B; no discounting; 8 episodes of experience

$A, 0, B, 0$

$B, 1$

$B, 1$

$B, 1$

$B, 1$

$B, 1$

$B, 1$

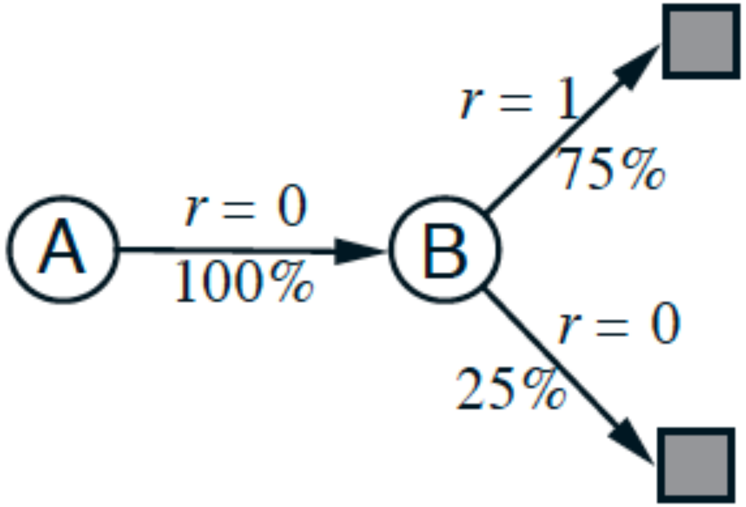
$B, 0$

What is $V(A), V(B)$?

AB Example

Two states A, B; no discounting; 8 episodes of experience

$A, 0, B, 0$
$B, 1$
$B, 1$
$B, 1$
$B, 1$
$B, 1$
$B, 1$
$B, 0$



What is $V(A), V(B)$? $V(B) = 0.75$

Batch MC

- MC converges to solution with minimum mean-squared error

- Best fit to the observed returns

$$\sum_{k=1}^{K} \sum_{t=0}^{T_k - 1} \left(G_t^k - V(s_t^k) \right)^2$$

- In the AB example, $V(A) = 0$

Batch TD(0)

- TD(0) converges to solution of max likelihood Markov model
- Solution to the MDP $\langle \mathcal{S}, \mathcal{A}, \hat{P}, \hat{r}, \gamma \rangle$ that best fits the data

$$\hat{P}_{ss'}(a) = \frac{1}{N(s,a)} \sum_{k=1}^{K} \sum_{t=0}^{T_k - 1} \mathbf{1}(s_t^k, a_t^k, s_{t+1}^k = s, a, s')$$

$$\hat{r}(s,a) = \frac{1}{N(s,a)} \sum_{k=1}^{K} \sum_{t=0}^{T_k - 1} \mathbf{1}\big(s_t^k, a_t^k = s, a\big) r_{t+1}^k$$

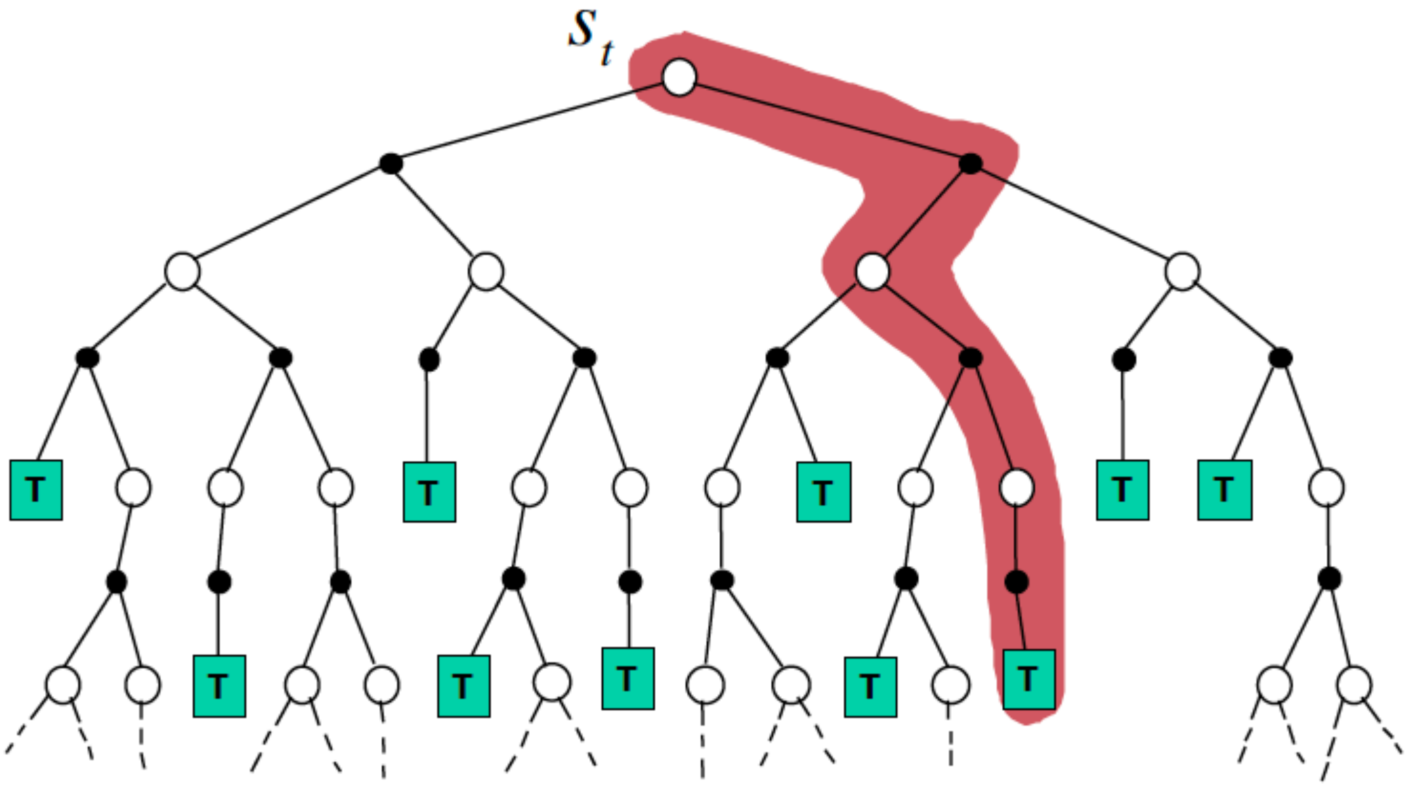
- Called *certainty-equivalence estimate*
- In the AB example, $V(A) = 0.75$

MC vs. TD (3)

- TD exploits Markov property
  - Usually more efficient in Markov environments
- MC does not exploit Markov property
  - Usually more efficient in non-Markov environments

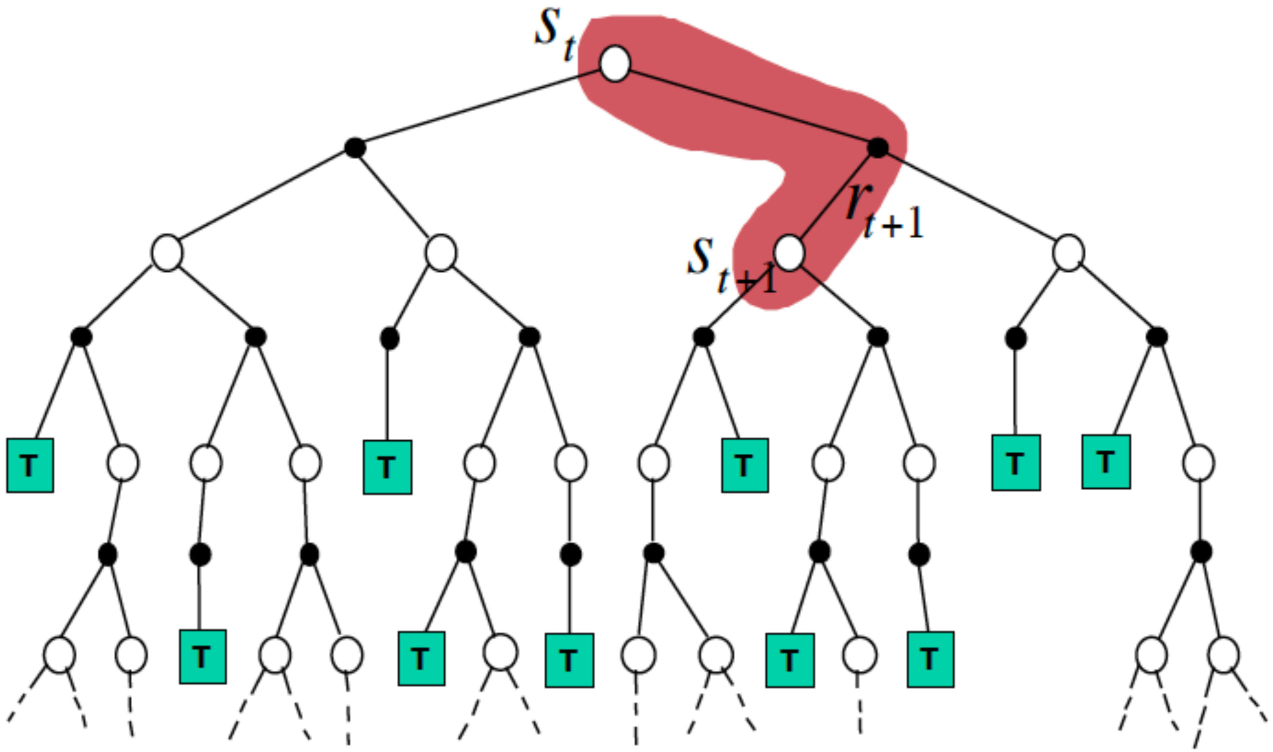
Monte-Carlo Backup

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



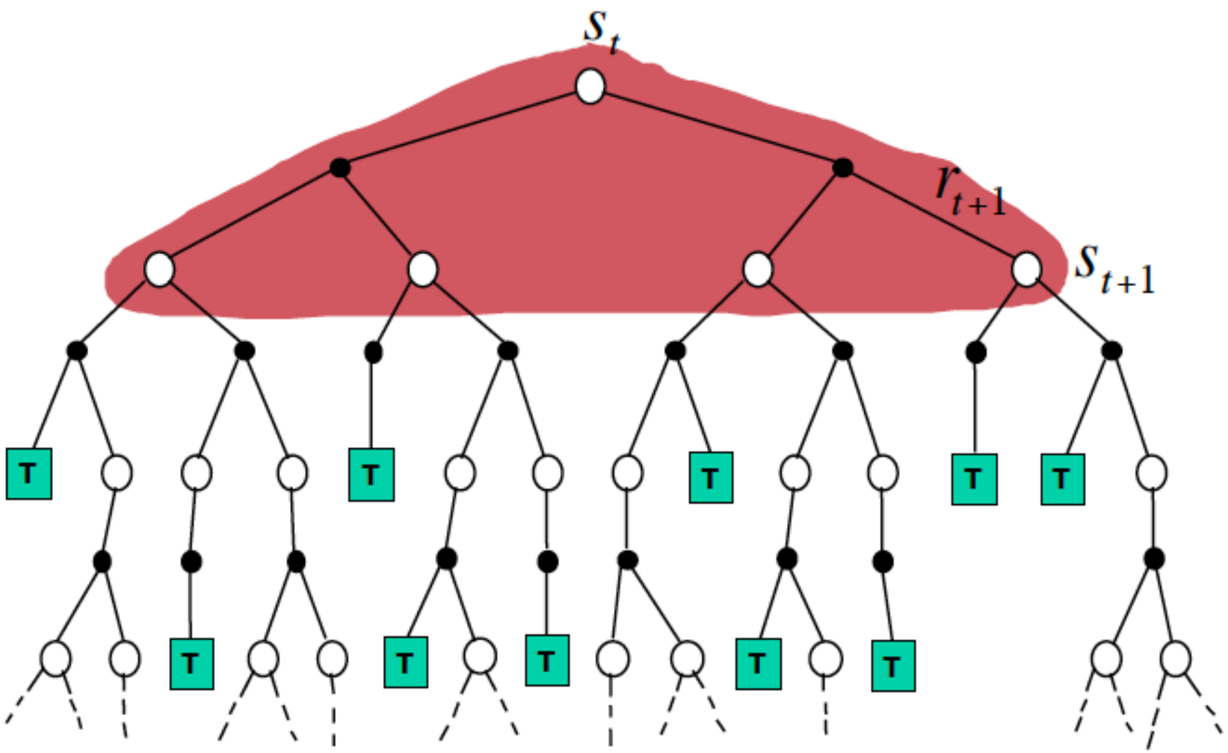
TD(0) Backup

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



Dynamic Programming Backup

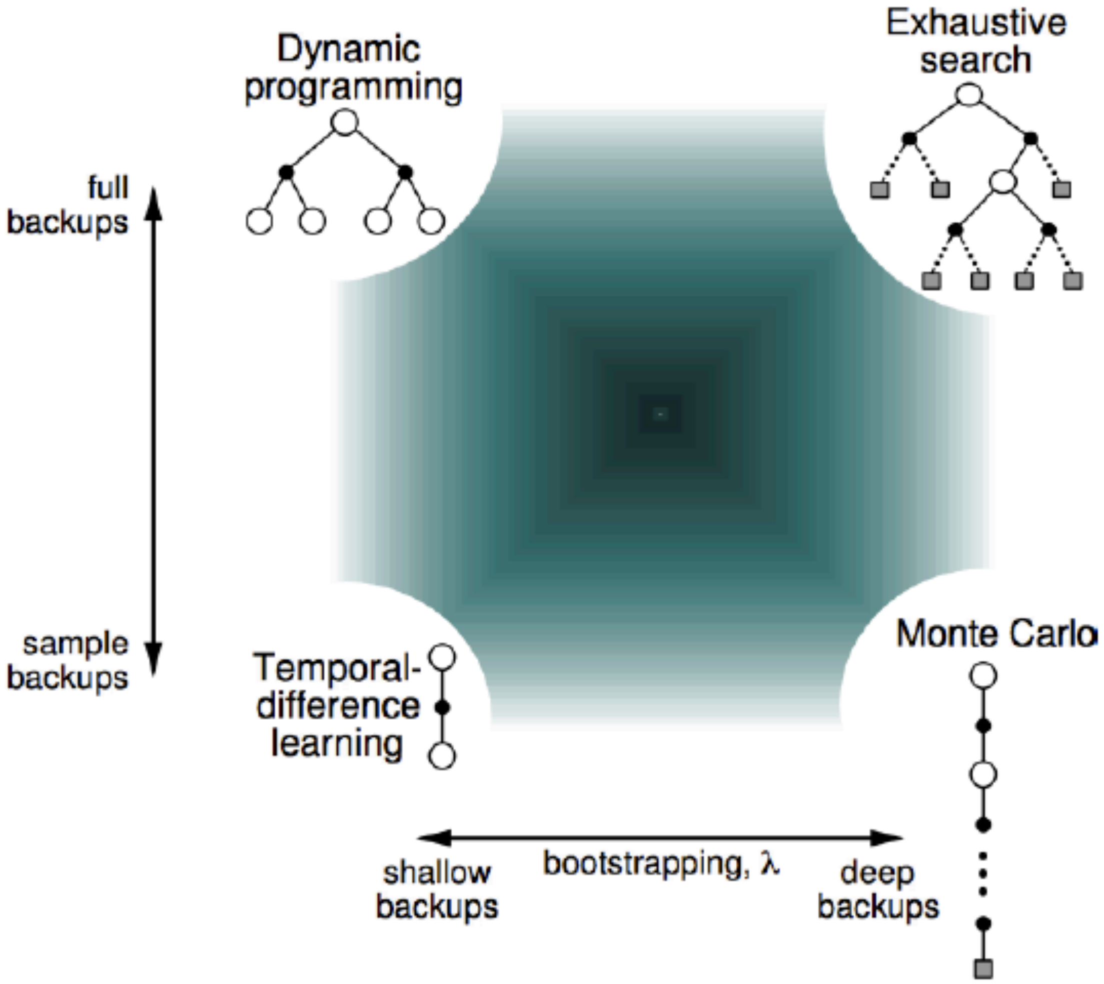
$$V(S_t) \leftarrow \mathrm{E}_\pi[R_{t+1} + \gamma V(S_{t+1})]$$



Bootstrapping and Sampling

- Bootstrapping: update involves an estimate
  - MC does not bootstrap
  - DP bootstraps
  - TD bootstraps
- Sampling: update samples an expectation
  - MC samples
  - DP does not sample
  - TD samples

Unified View of Reinforcement Learning



Agenda

- Monte Carlo Method
- TD(0)
- n-step TD
- TD(λ)

Expressions of Value Function

- Conditional expectation of return:

$$v_\pi(s) = \mathbb{E}_\pi(G_t|S_t = s)$$

- Bellman Equation:

$$v_\pi(s) = \mathbb{E}_\pi(R_{t+1} + \gamma\, v_\pi(S_{t+1})|S_t = s)$$
$$v_\pi(s) = \mathbb{E}_\pi(R_{t+1} + \gamma\, R_{t+2} + \gamma^2\, v_\pi(S_{t+2})|S_t = s)$$
$$v_\pi(s) = \mathbb{E}_\pi(R_{t+1} + \gamma\, R_{t+2} + \gamma^2\, R_{t+3} + \gamma^3\, v_\pi(S_{t+3})|S_t = s)$$

…

n-Step Return

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

$$n = 1 \quad \text{(TD(0))} \qquad G_t^{(1)} = R_{t+1} + \gamma V(S_{t+1})$$

$$n = 2 \qquad G_t^{(2)} = R_{t+1} + \gamma R_{t+2} + \gamma^2 V(S_{t+2})$$

$$\vdots \qquad \vdots$$

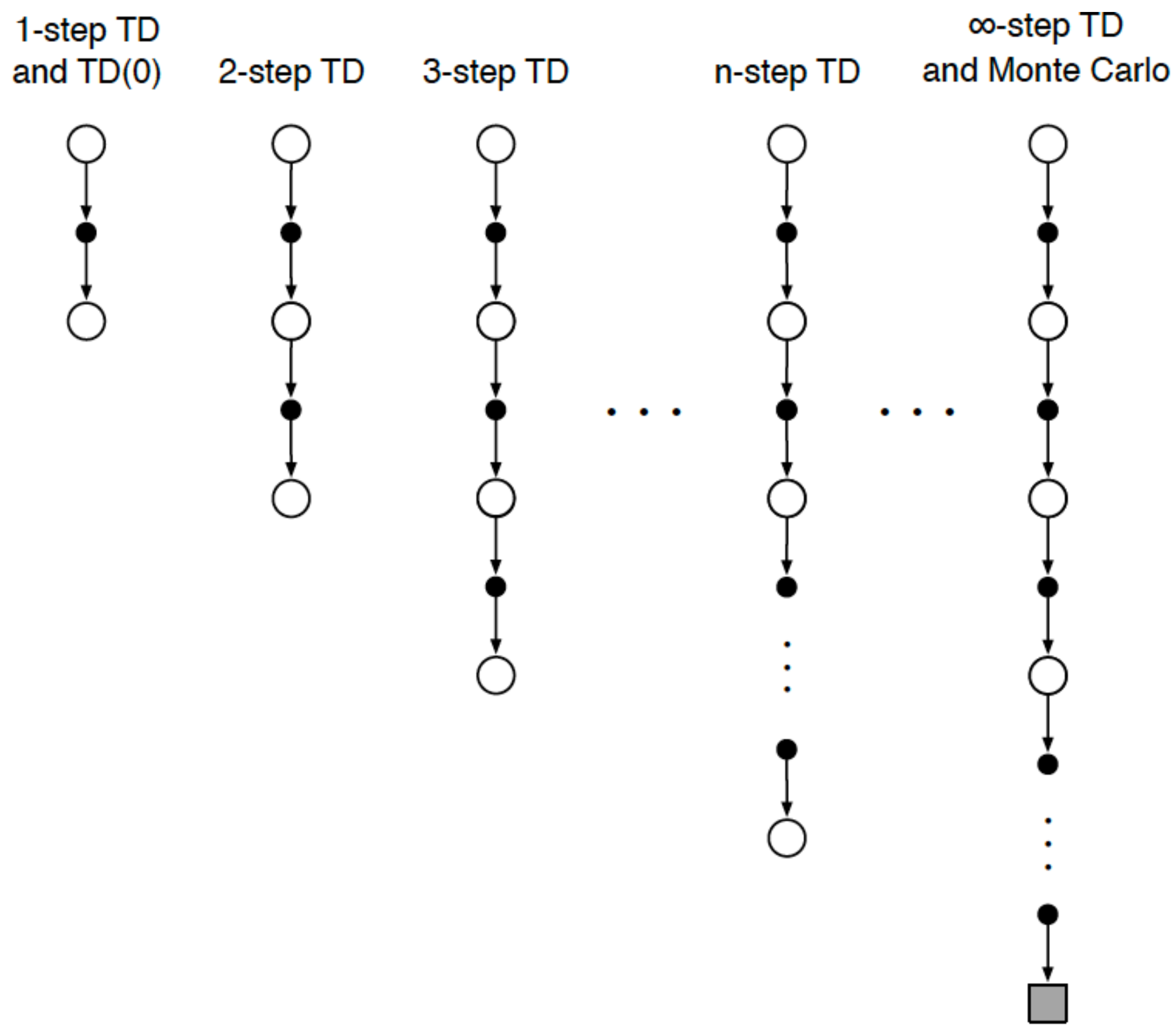
$$n = \infty \quad \text{(MC)} \qquad G_t^{(\infty)} = R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{T-t-1} R_T$$

- $G_t^{(T-t-1)} = G_t^{(T-t)} = \cdots = G_t^{(\infty)}$

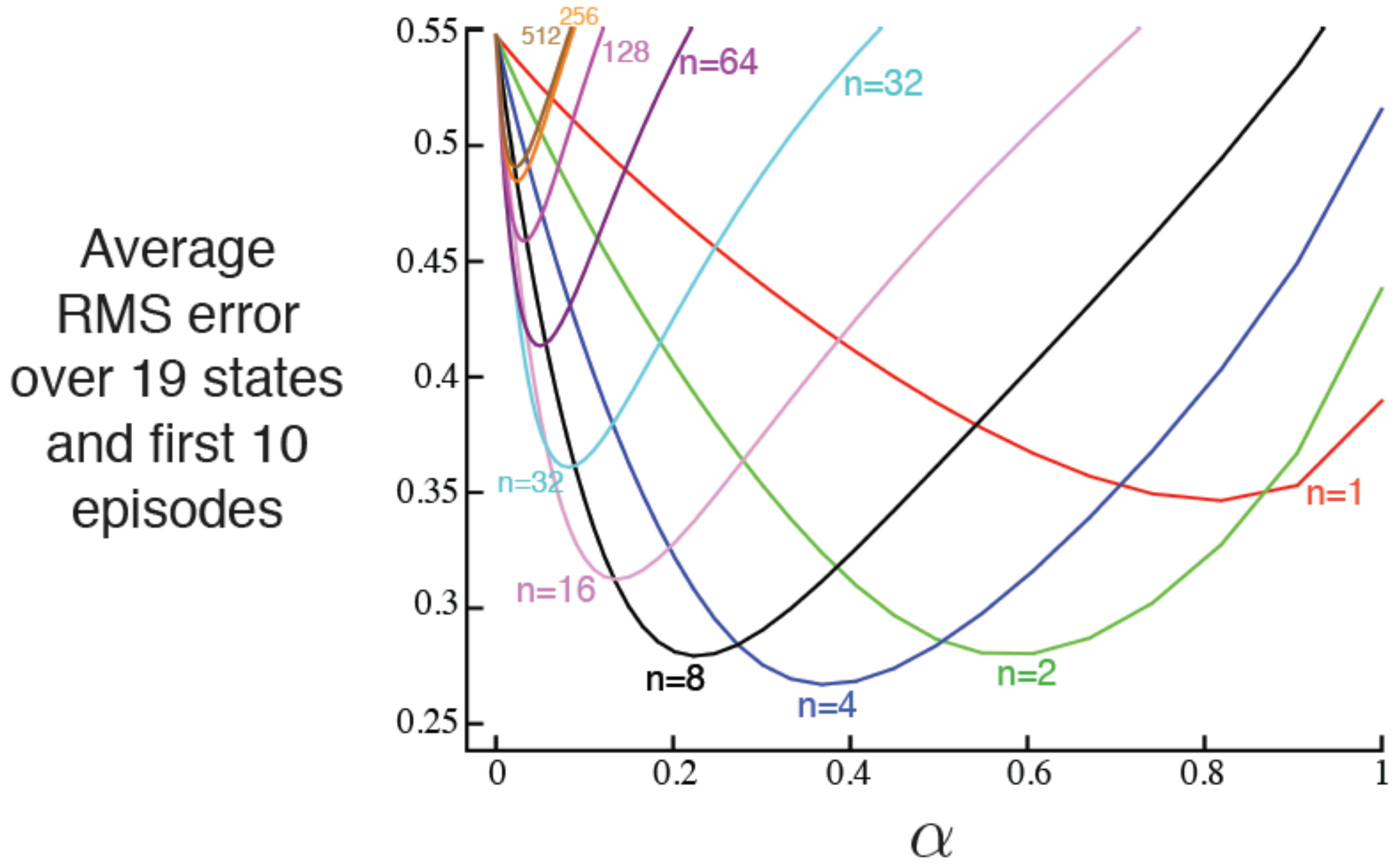
- n-step temporal-difference learning

$$V(S_t) = V(S_t) + \alpha \left(G_t^{(n)} - V(S_t) \right)$$

n-step TD


1-step TD
and TD(0)
2-step TD
3-step TD
n-step TD
∞-step TD
and Monte Carlo

Large Random Walk Example



Performance of n-step TD methods as a function of α, for various values of n, on a 19-state random walk task (Example 7.1 in SB).

Agenda

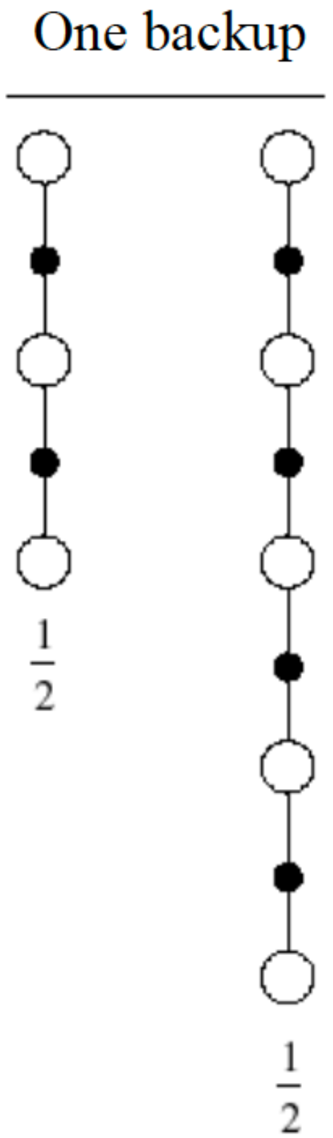
- Monte Carlo Method
- TD(0)
- n-step TD
- TD(λ)

Averaging n-Step Returns

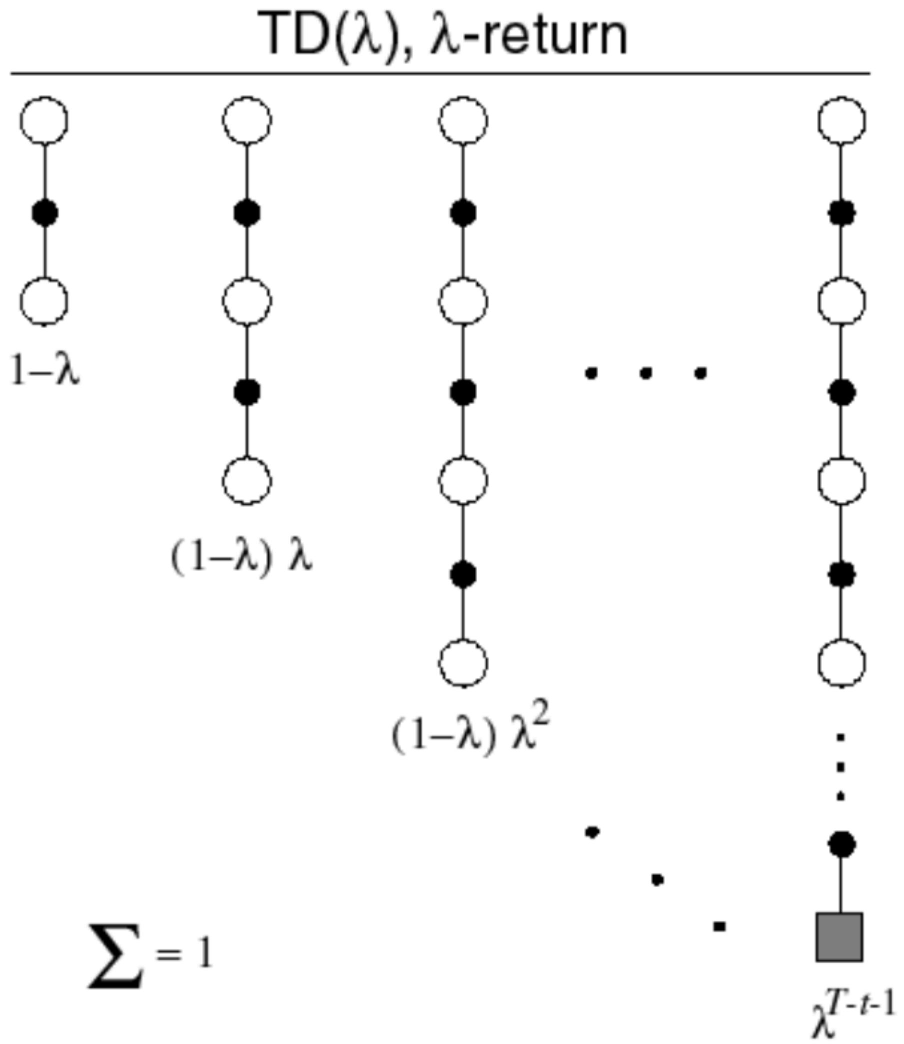
- We can average n-step returns over different n
- e.g. average the 2-step and 4-step returns

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

- Combines information from two different time-steps
- Can we efficiently combine information from all time-steps?



λ-return



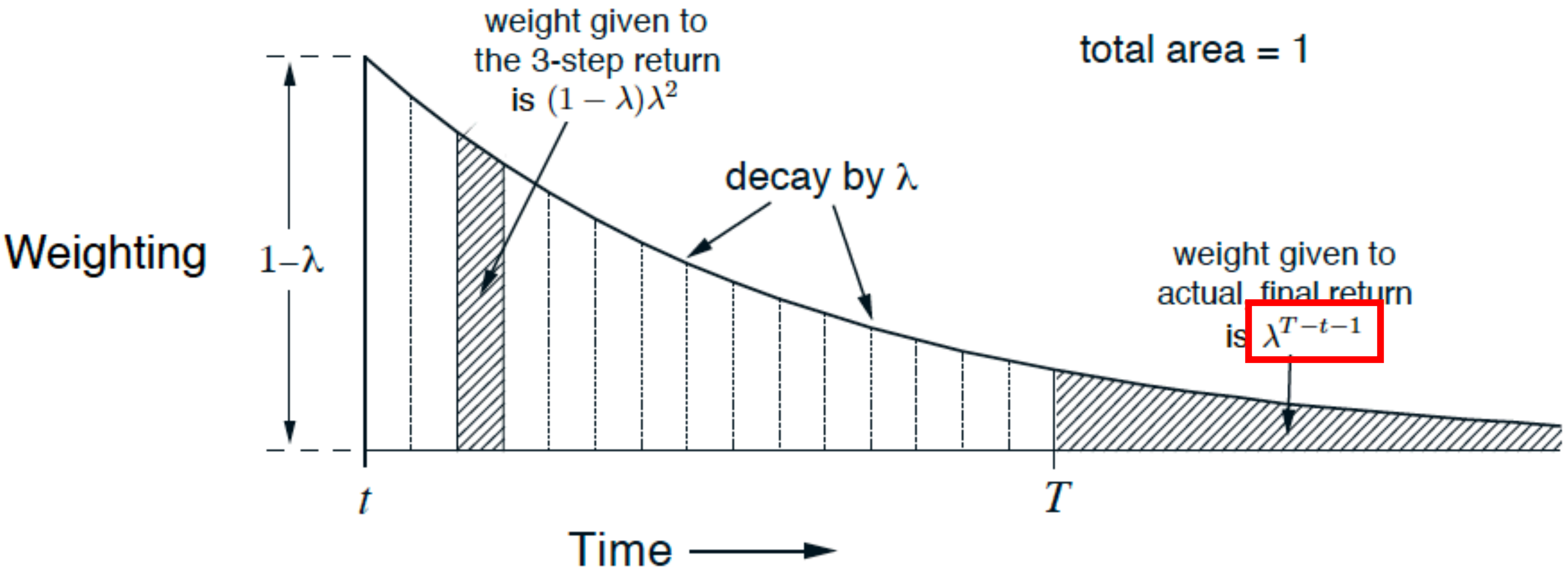
- The λ-return G_t^λ combines all n-step return $G_t^{(n)}$
- Using weight $(1-\lambda)\lambda^{n-1}$

$$G_t^\lambda \doteq (1-\lambda)\sum_{n=1}^{\infty}\lambda^{n-1}G_t^{(n)}$$

- Forward-view TD(λ)

$$V(S_t) = V(S_t) + \alpha\left(G_t^\lambda - V(S_t)\right)$$

TD(λ) weighting function



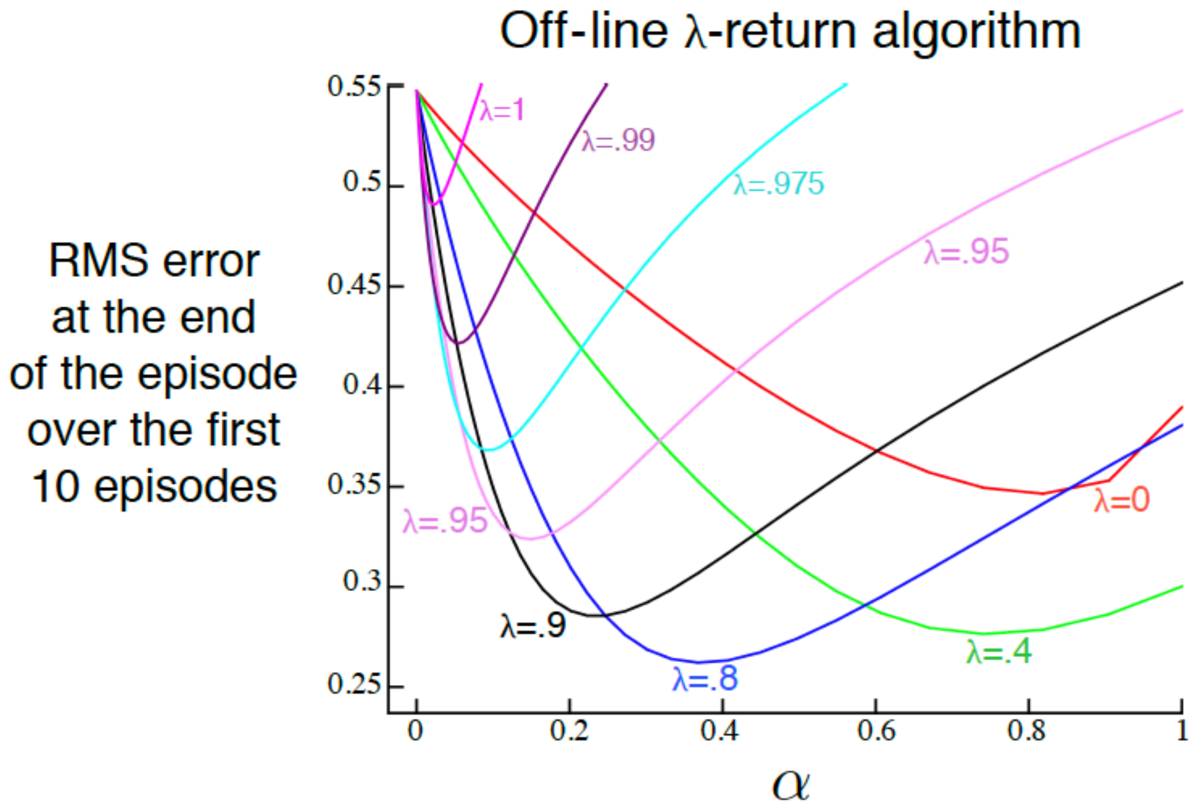
$$G_t^\lambda = (1-\lambda)\sum_{n=1}^{\infty}\lambda^{n-1}G_t^{(n)}$$

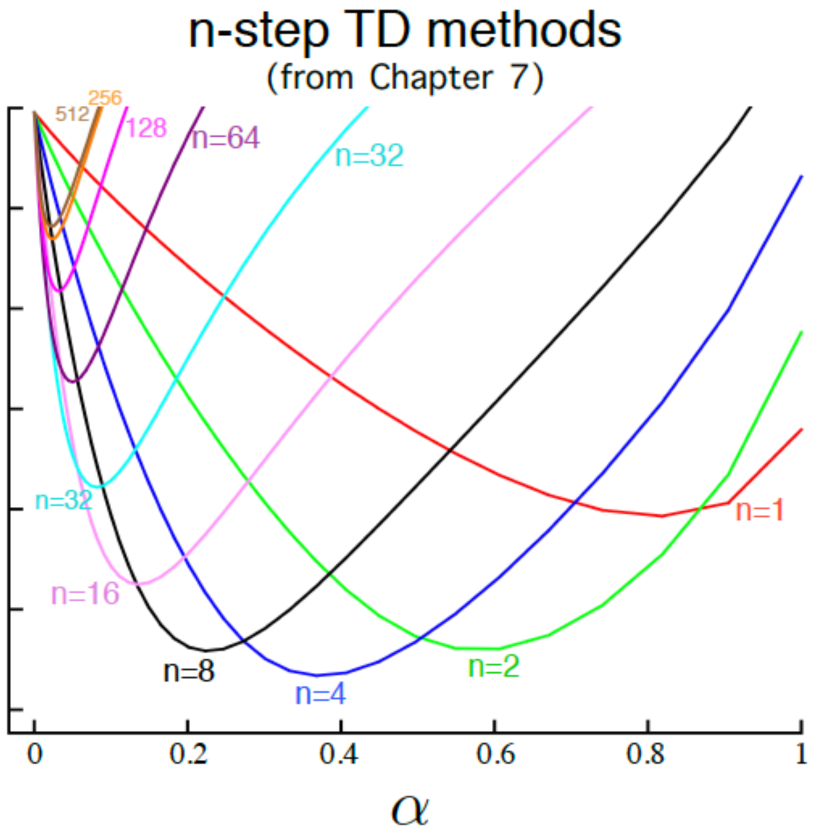
$$= (1-\lambda)\sum_{n=1}^{T-t-1}\lambda^{n-1}G_t^{(n)} + \lambda^{T-t-1}G_t$$

$\Rightarrow G_t^\lambda = G_t^{(1)}$ when $\lambda = 0$

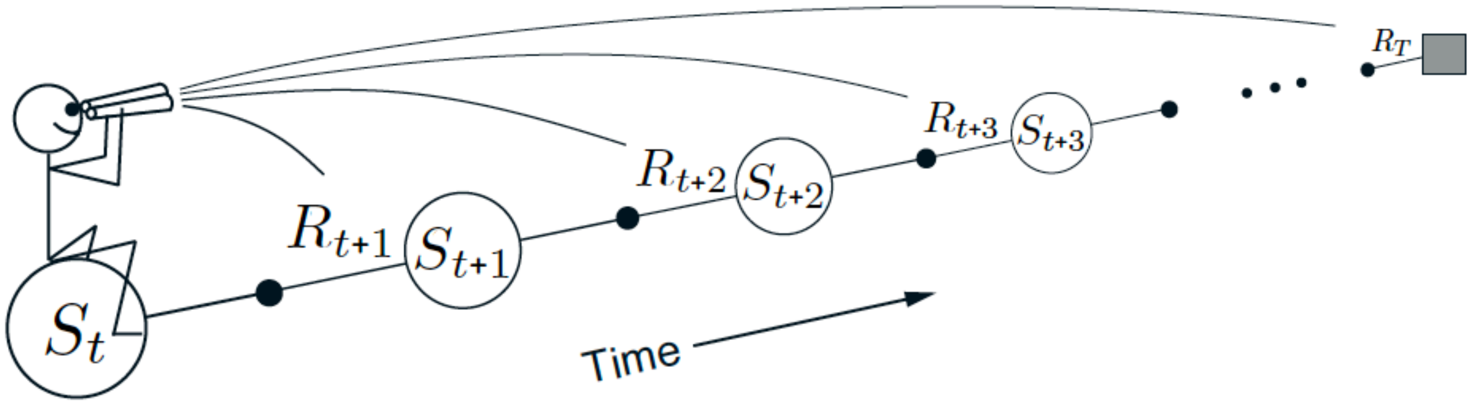
$G_t^\lambda = G_t$ when $\lambda = 1$

Forward-View TD(λ) on Large Random Walk





Forward View TD(λ)



- Update value function towards the λ-return
- Forward-view looks into the future to compute G_t^λ
- Like MC, can only be computed from complete episodes

Backward View TD(λ)

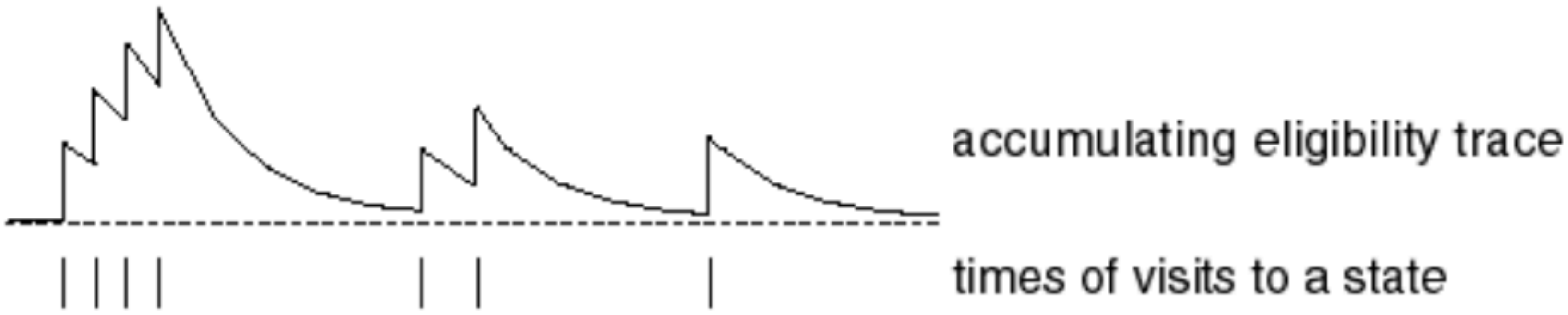
- Forward view provides theory
- Backward view provides mechanism
- Update online, every step, from incomplete sequences

Eligibility Traces

- Frequency heuristic: assign credit to most frequent states
- Recency heuristic: assign credit to most recent states
- Eligibility traces combine both heuristics

$$E_{-1}(s) = 0$$

$$E_t(s) = \gamma\lambda E_{t-1}(s) + \mathbf{1}(S_t = s)$$

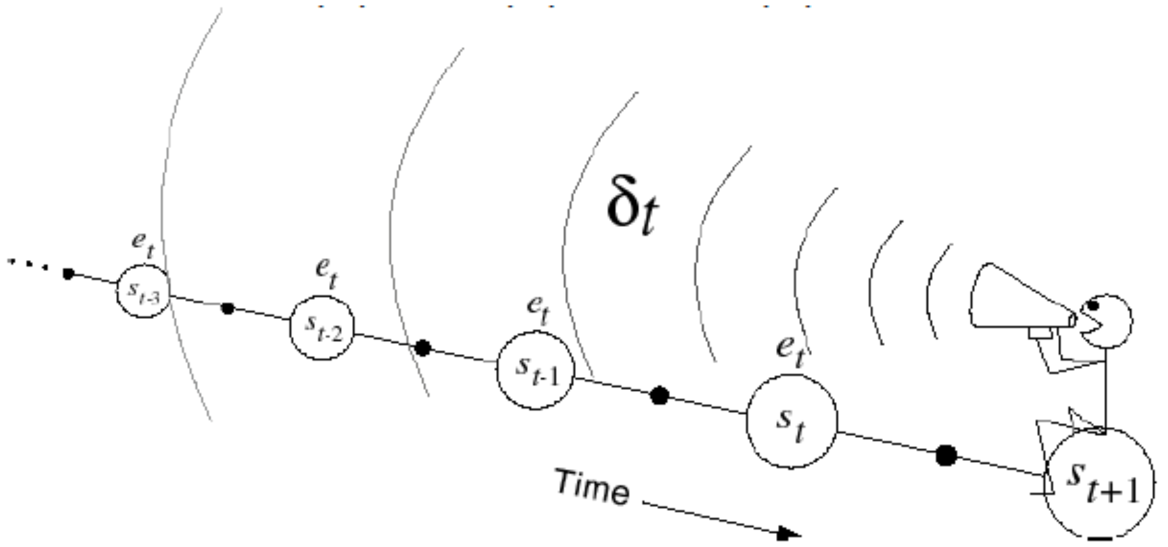


Backward View TD(λ)

- Keep an eligibility trace for every state s
- Update value $V(s)$ for every state s
- In proportion to TD-error δ_t and eligibility trace $E_t(s)$

$$\delta_t = R_{t+1} + \gamma V(S_{t+1}) - V(S_t)$$

$$V(s) = V(s) + \alpha \delta_t E_t(s), \forall s \in \mathcal{S}$$



Tabular TD(λ) for estimating v_π

Input: π: policy to be evaluated, α: step size, $\lambda \in [0,1]$: trace decay rate

Initialize $V(s)$ for $s \in \mathcal{S}^+$, arbitrarily except $V(s^*) = 0$

Loop for each episode:

  Initizlize S, $E(s) = 0, \forall s$

  Loop for each step of epsiode:

    Choose $A \sim \pi(\cdot\,|S)$

    Take action A, observe R, S'

    $E(s) \leftarrow \gamma\lambda E(s) + \mathbf{1}(S = s), \forall s$

    $\delta = R + \gamma V(S') - V(S)$

    $V(s) \leftarrow V(s) + \alpha\delta E(s), \forall s$

    $S \leftarrow S'$

  until S is terminal

TD(λ) and TD(0)

- When $\lambda = 0$, only current state is updated

$$E_t(s) = \mathbf{1}(S_t = s)$$

$$V(s) = V(s) + \alpha\delta_t E_t(s)$$

- This is exactly equivalent to TD(0) update

$$V(S_t) = V(S_t) + \alpha\delta_t$$

Offline Equivalence of Forward and Backward TD

Offline updates

- Updates are accumulated within episode
- but applied in batch at the end of episode

Theorem

The sum of offline updates is identical for forward-view and backward-view $TD(\lambda)$

$$\sum_{t=0}^{T-1} \alpha \delta_t E_t(s) = \sum_{t=0}^{T-1} \alpha \left(G_t^{\lambda} - V(S_t)\right) \mathbf{1}(S_t = s), \forall s \in \mathcal{S}$$

TD(1) and MC

- TD(1) is roughly equivalent to every-visit Monte-Carlo
- When $\lambda = 1$, credit is deferred until end of episode
- Consider episodic environments with offline updates
- Error is accumulated online, step-by-step
- If value function is only updated offline at the end of episode
- Then total update is exactly the same as MC

TD(1) and MC

- Consider an episode where s is visited once at time-step k,
- TD(1) eligibility trace discounts time since visit,

$$E_t(s) = \gamma E_{t-1}(s) + \mathbf{1}(S_t = s)$$

$$= \begin{cases} 0 & \text{if } t < k \\ \gamma^{t-k} & \text{if } t \geq k \end{cases}$$

- TD(1) updates accumulate error *online*

$$\sum_{t=0}^{T-1} \alpha \delta_t E_t(s) = \alpha \sum_{t=k}^{T-1} \gamma^{t-k} \delta_t$$

$$= \alpha\left(\delta_k + \gamma \delta_{k+1} + \cdots + \gamma^{T-1-k} \delta_{T-1}\right)$$

$$= \alpha\left(G_k - V(S_k)\right)$$

Telescoping in TD(1)

$$
\begin{aligned}
&\delta_t + \gamma\delta_{t+1} + \cdots + \gamma^{T-1-t}\delta_{T-1} \\
&= R_{t+1} + \gamma V(S_{t+1}) - V(S_t) \\
&\quad +\gamma R_{t+2} + \gamma^2 V(S_{t+2}) - \gamma V(S_{t+1}) \\
&\quad +\gamma^2 R_{t+3} + \gamma^3 V(S_{t+3}) - \gamma^2 V(S_{t+2}) \\
&\qquad \vdots \\
&\quad +\gamma^{T-1-t} R_T + \gamma^{T-t} V(S_T) - \gamma^{T-t-1} V(S_{T-1}) \\
&= R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{T-1-t} R_T - V(S_t) \\
&= G_t - V(S_t)
\end{aligned}
$$

When $\lambda = 1$, sum of TD errors telescopes into MC error

Online Equivalence of Forward and Backward TD

Online updates

- TD(λ) updates are applied online at each step within episode
- Forward and backward-view TD(λ) are slightly different
- NEW: Exact online TD(λ) achieves perfect equivalence
- By using a slightly different form of eligibility trace
- Sutton and von Seijen, ICML 2014

Some evidence for TD

- Psychology recognizes two fundamental learning processes, analogous to prediction and control

- The details of the TD(λ) algorithm match key features of biological learning
  - Dopamine = TD error is the most important interaction ever between AI and neuroscience

- Read SB 15.6