Model-free Prediction

Weizi Li

Department of Computer Science
University of Memphis

- Introduction
- Monte-Carlo Prediction
- Temporal-Difference Prediction

Introduction

- Model-based prediction and control: solve a *known* MDP (the transition function and reward function are known)

- Model-free prediction (evaluate a given policy): estimate the value function of an *unknown* MDP
  - Monte-Carlo Prediction
  - Temporal-Difference Prediction
- Model-free control (find the best policy, next lecture): optimize the value function of an *unknown* MDP

Monte-Carlo Prediction

- Learns from experience: sample sequences of states, action, and rewards from actual or simulated interactions with the environment
- Do not need to know the reward function and transition function of the MDP (i.e., *unknown* MDP)
- Learns from *complete* episodes: can only be applied to *episodic* MDPs (i.e., all episodes will terminate)

- The goal is to compute V^{π} from episodes of experience under π

$$S_1, A_1, R_1, ..., S_T \sim \pi$$

- Recall the definition of return

$$G_t \equiv R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \cdots + \gamma^T R_T$$

and value function

$$V^{\pi}(s) \equiv \mathbb{E}_{\pi}[G_t | S_t = s]$$

- MC policy evaluation uses *empirical mean* return instead of *expected* return

- Goal: evaluate state s under π (compute $V^{\pi}(s)$)
- The *first* timestep t that state s is visited in an episode:
  - increment counter $N(s) \leftarrow N(s) + 1$
  - increment total return $S(s) \leftarrow S(s) + G_t$
- Estimate value using the average return $V(s) = \frac{S(s)}{N(s)}$, by law of large numbers, $V(s) \rightarrow V^{\pi}(s)$ as $N(s) \rightarrow \infty$

Every-visit MC Policy Evaluation

- Goal: evaluate state s under π (compute $V^\pi(s)$)
- *Every* timestep t that state s is visited in an episode:
  - increment counter $N(s) \leftarrow N(s) + 1$
  - increment total return $S(s) \leftarrow S(s) + G_t$
- Estimate value using the average return $V(s) = \frac{S(s)}{N(s)}$, by law of large numbers, $V(s) \rightarrow V^\pi(s)$ as $N(s) \rightarrow \infty$

First-visit MC vs. Every-visit MC

- First-visit MC estimator is an *unbiased* estimator of V^π while every-visit MC is *biased*: the former uses i.i.d. estimates and the latter uses non-i.i.d. estimates (the visit counts are correlated).
- Both first-visit MC and every-visit MC are *consistent estimator* meaning given enough samples, they will converge to the true values.
- Empirically, every-visit MC has lower variance and can outperform first-visit MC due to the use of more samples.

$$
\begin{aligned}
\mu_k &= \frac{1}{k}\sum_{j=1}^{k} x_j \\
&= \frac{1}{k}\left(x_k + \sum_{j=1}^{k-1} x_j\right) \\
&= \frac{1}{k}\left(x_k + (k-1)\,\mu_{k-1}\right) \\
&= \mu_{k-1} + \frac{1}{k}(x_k - \mu_{k-1})
\end{aligned}
$$

- Update $V(s)$ incrementally after each episode:

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

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

- For non-stationary problems, we can track a running mean:

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

Temporal-Difference Prediction

- “If one had to identify one idea as central and novel to reinforcement learning, it would undoubtedly be temporal-difference (TD) learning.”—Sutton and Barto

- Similarity to MC: learns from experience; works for unknown MDP
- Difference to MC: learns from *incomplete* episodes (even in infinite-horizon settings) using *bootstrapping* (updates a guess using a guess)

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

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

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

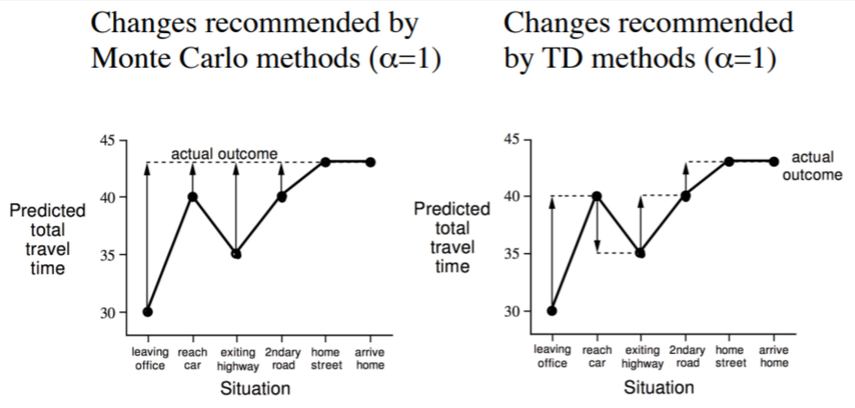
Tabular TD(0) for estimating v_π

Input: 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 $V(\textit{terminal}) = 0$

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

State	Elapsed Time (minutes)	Predicted Time to Go	Predicted Total Time
leaving office	0	30	30
reach car, raining	5	35	40
exit highway	20	15	35
behind truck	30	10	40
home street	40	3	43
arrive home	43	0	43

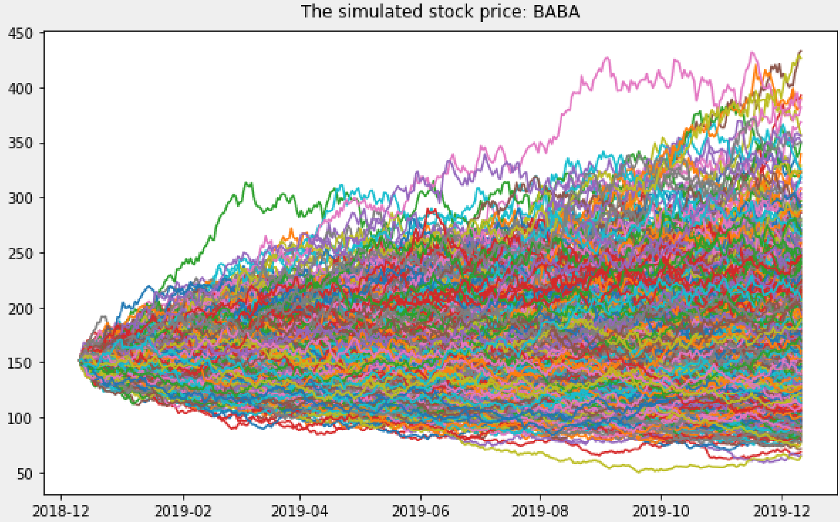
Time to Home: MC vs. TD(0)

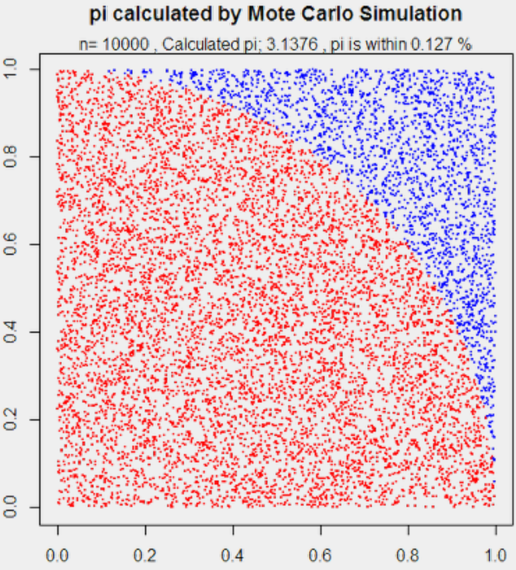


- MC must wait for the return from complete sequences.
- MC only works for episodic (terminating) environments.
- TD can learn online after every step, from incomplete sequences.
- TD can work in continuing (non-terminating) environments.

- Both the return and true TD target are *unbiased* estimate of the value function.
- However, the actual TD target is often *biased* estimate of the value function due to the use of biased estimates.

- TD target has much lower variance than return:
  - Return depends on *many* random actions, transitions, and rewards.
  - TD target depends on *one* random action, transition, and reward.


The simulated stock price: BABA
450
400
350
300
250
200
150
100
50
2018-12
2019-02
2019-04
2019-06
2019-08
2019-10
2019-12


pi calculated by Mote Carlo Simulation
n= 10000 , Calculated pi; 3.1376 , pi is within 0.127 %
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0

- MC has high variance, zero bias
  - Good convergence properties (even with function approximation)
  - Not very sensitive to the initial value of V
  - Easy to understand and use
- TD has low variance, some bias
  - TD(0) converges to $V^{\pi}(s)$ (but not always with function approximation)
  - More sensitive to the initial value of V
  - Usually more efficient than MC

- MC and TD converge to V^{π} when the number of experiences $\to \infty$
- In practice, we have a finite number of experiences (a batch)

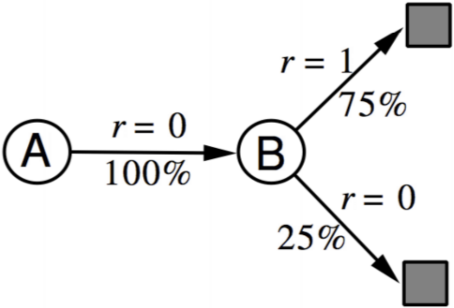
Two states A, B; no discounting; 8 episodes of experience

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

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

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

- MC: $V(A) = 0$, $V(B) = 0.75$
- TD: $V(A) = 0.75$, $V(B) = 0.75$

- MC converges to solution with minimum mean-squared error
  - Best fit to the observed returns

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

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

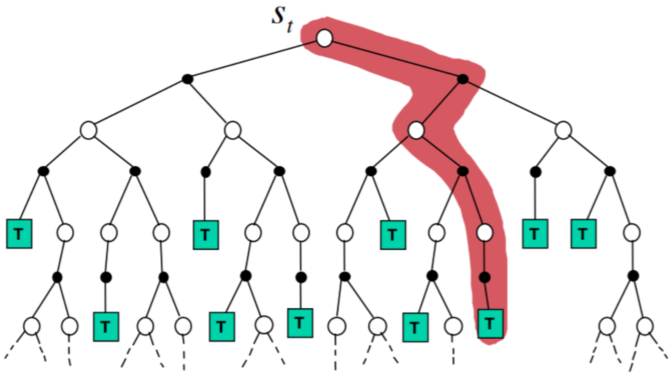
$$\hat{\mathcal{P}}_{s,s'}^a = \frac{1}{N(s,a)} \sum_{k=1}^{K}\sum_{t=1}^{T_k} \mathbf{1}(s_t^k, a_t^k, s_{t+1}^k = s, a, s')$$

$$\hat{\mathcal{R}}_s^a = \frac{1}{N(s,a)} \sum_{k=1}^{K}\sum_{t=1}^{T_k} \mathbf{1}(s_t^k, a_t^k = s, a) r_t^k$$

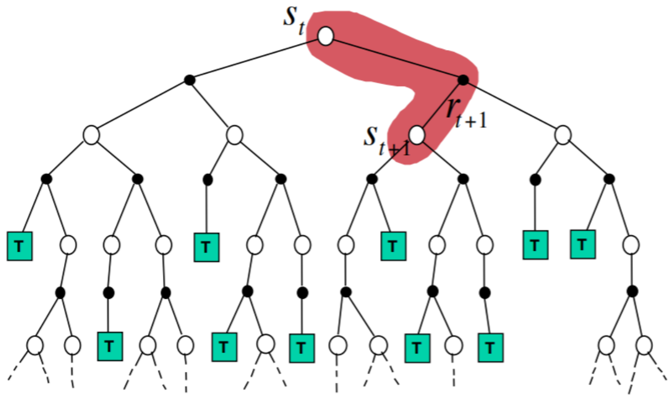
  - In the AB example, $V(A) = 0.75$

- All are *consistent estimator* and converge to true value functions when using tabular representations, though only MC is unbiased.
- With function approximation, MC can still converge but TD may fail.
- DP needs model; MC or TD does not.
- DP and TD can work under continuing (non-episodic) settings; MC can't.
- DP and TD exploit the Markov property; MC doesn't.

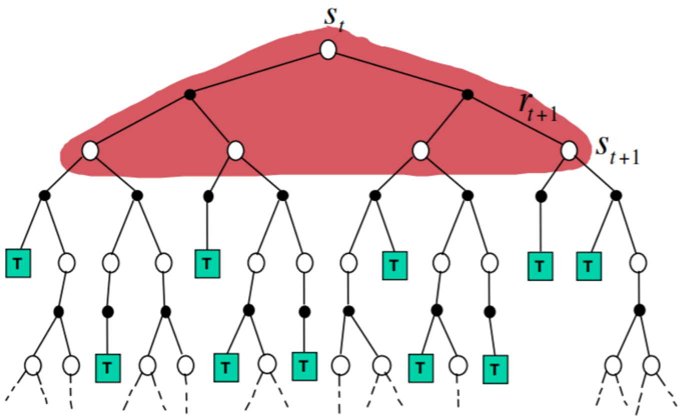
$$V(S_t) \leftarrow V(S_t) + \alpha\left(G_t - V(S_t)\right)$$



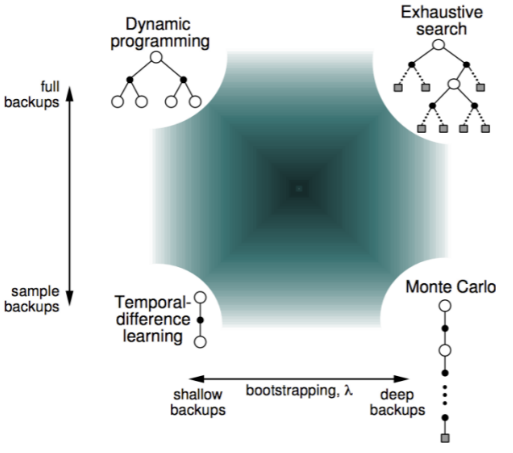
$$V(S_t) \leftarrow V(S_t) + \alpha \left(R_{t+1} + \gamma V(S_{t+1}) - V(S_t) \right)$$



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

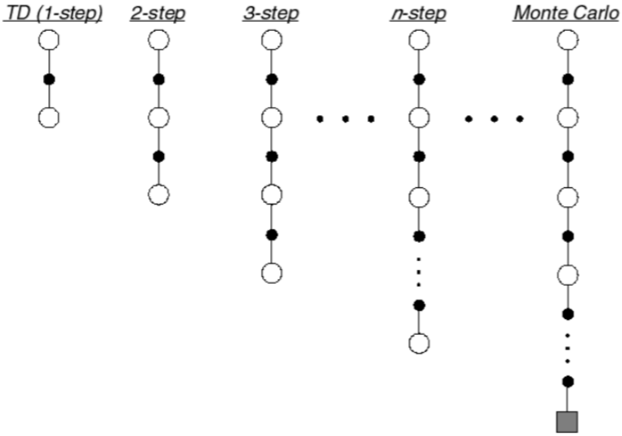


- Bootstrapping: update via estimates
- Sampling: update via sampled experiences



n-Step Prediction

- Let TD target look *n* steps into the future



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

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

- Define the n-step return

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

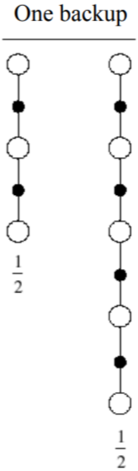
- n-step temporal-difference learning

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

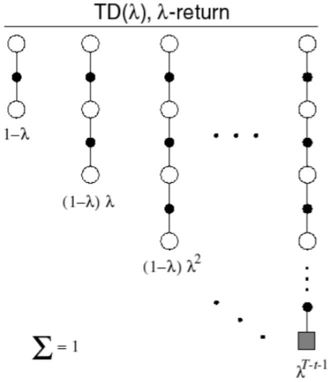
- We can average n-step returns over different n
- e.g. average the 2-step and 4-step returns

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

- 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 returns $G_t^{(n)}$
- Using weight $(1-\lambda)\lambda^{n-1}$

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

- Forward-view TD(λ)

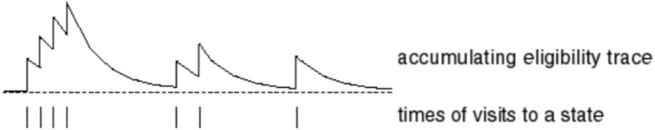
$$V(S_t) \leftarrow V(S_t) + \alpha \left(G_t^\lambda - V(S_t) \right)$$

- Forward-view
  - Update the value function towards the λ-return
  - Looks into the future to compute G_t^λ
  - Similar to MC, can only be computed from complete episodes
- Backward-view
  - Provides a mechanism to update online, at every step, from incomplete sequences

Eligibility Traces

- Credit assignment problem: did bell or light cause shock?
- Frequency heuristic: assign credit to most frequent states
- Recency heuristic: assign credit to most recent states
- *Eligibility traces* combine both heuristics

$$E_0(s) = 0$$
$$E_t(s) = \gamma\lambda E_{t-1}(s) + \mathbf{1}(S_t = s)$$



- Keep an eligibility trace for every state s
- Update $V(s)$ for all s, in proportion to TD-error δ_t and eligibility trace $E_t(s)$:

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

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

- When $\lambda = 0$, we have TD(0), only the current state is updated

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

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

- When $\lambda = 1$, we have MC, credit is deferred until the end of episode