This lecture will be recorded!!!

Welcome to

DS595 Reinforcement Learning

Prof. Yanhua Li

Time: 6:00pm –8:50pm W
Zoom Lecture
Spring 2022

Happy Lunar New Year

Last lecture

- Reinforcement Learning Components
  - Model, Value function, Policy
- Model-based Control
  - Policy Evaluation, Policy Iteration, Value Iteration
- Project 1 description.

Quiz 1 Week 4 (2/9 W)

- Model-based Control
  - Policy Evaluation, Policy Iteration, Value Iteration
  - 30 min at the beginning on 2/9 W Week #4
    - You can start as early as 5:55PM, and finish as late as 6:30PM. The quiz duration is 30 minutes.
  - Login class zoom so you can ask questions regarding the quiz in Zoom Chatbox.

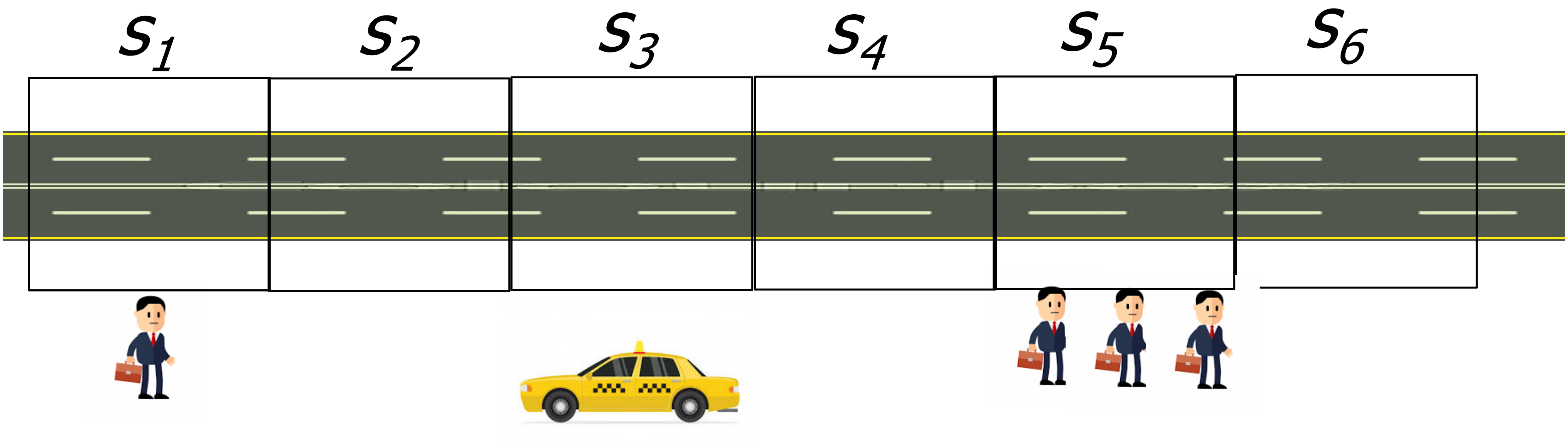
Project 1 due Week 4 (2/9 W)

This lecture

- Review: Model based control
  - Policy Iteration, and Value iteration

- Model-Free Policy Evaluation
  - Monte Carlo policy evaluation
  - Temporal-difference (TD) policy evaluation

Example: Taxi passenger-seeking task as a decision-making process



States: Locations of taxi ($s_1, \ldots, s_6$) on the road

Actions: Left or Right

Rewards:

+1 in state s_1

+3 in state s_5

0 in all other states

RL components

- Often include one or more of
  - **Model:** Representation of how the world changes in response to agent’s action
  - **Policy:** function mapping agent’s states to action
  - **Value function:** Future rewards from being in a state and/or action when following a particular policy

RL components: (1) Model

- Agent's representation of how the world changes in response to agent's action, with two parts:

Transition model
predicts next agent state

$p(s_{t+1} = s' \mid s_t = s, a_t = a)$

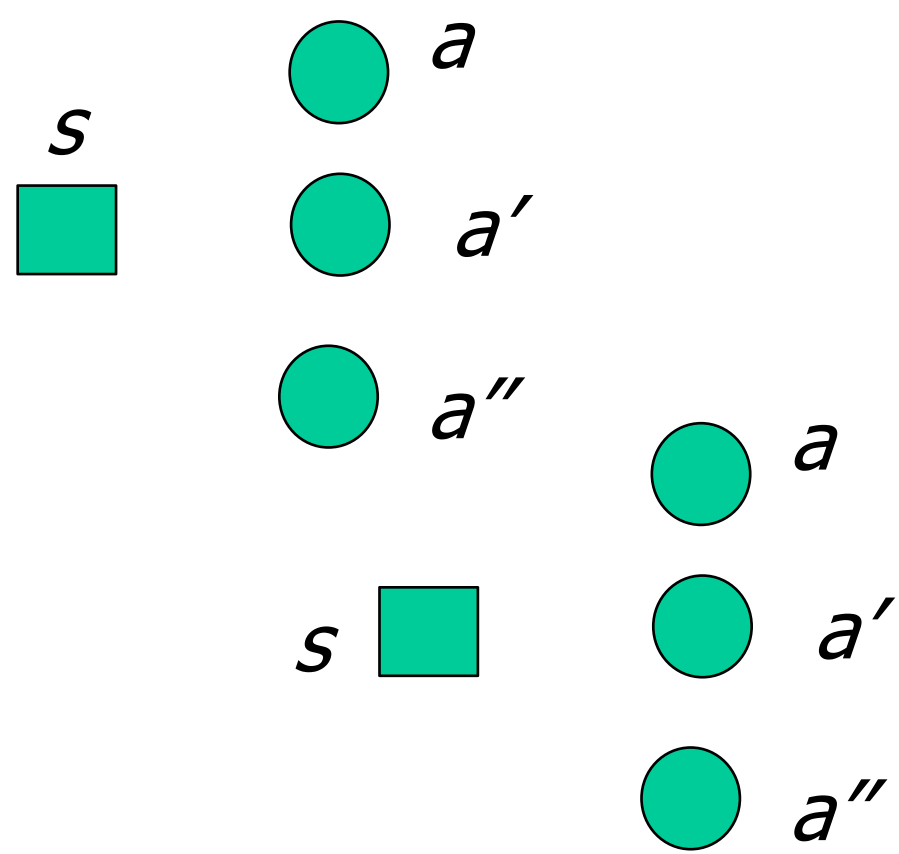
Reward model
predicts immediate reward

$r(s,a)$

RL components: (2) Policy

- Policy π determines how the agent chooses actions
  - $\pi : S \rightarrow A$, mapping from states to actions

- Deterministic policy:
  - $\pi(s) = a$
  - *In the other word,*
    - $\pi(a|s) = 1$,
    - $\pi(a'|s) = \pi(a''|s) = 0$,
- Stochastic policy:
  - $\pi(a|s) = \Pr(a_t = a|s_t = s)$

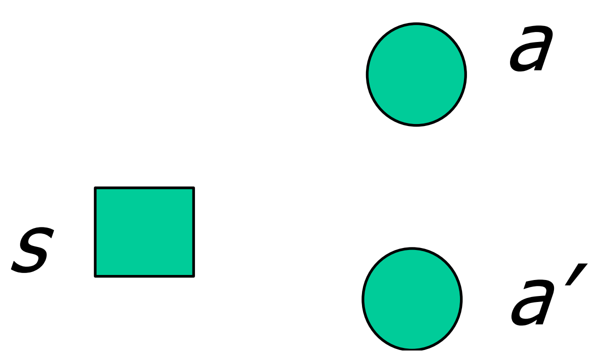


RL components: (3) Value Function

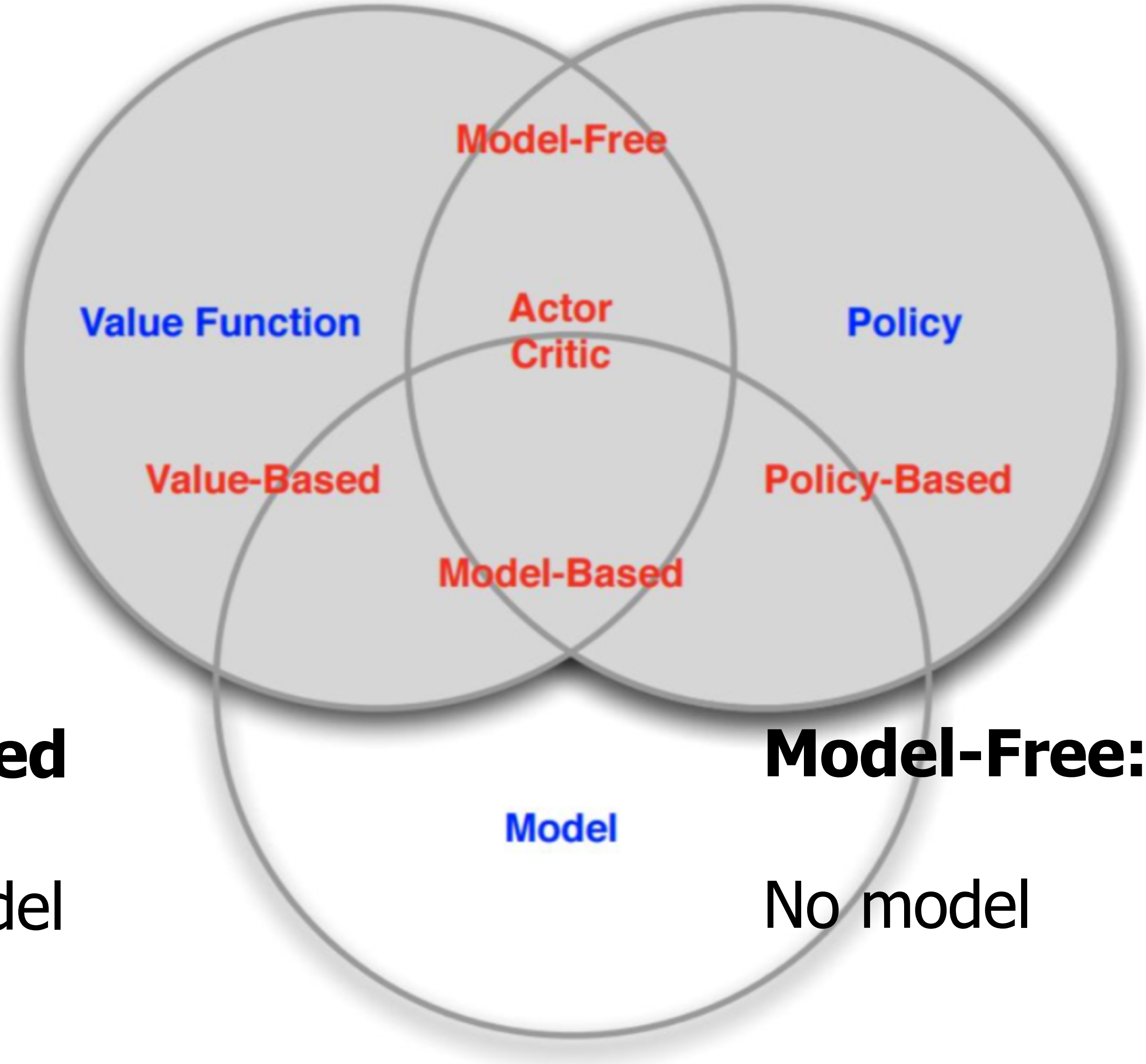
- Value function V^{π}: expected discounted sum of future rewards under a particular policy π

$$V^{\pi}(s_t = s) = \mathbb{E}_{\pi}[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \gamma^3 r_{t+3} + \cdots | s_t = s]$$

- Discount factor γ weighs immediate vs future rewards
- Can be used to quantify goodness/badness of states and actions
- And decide how to act by comparing policies



RL agents and algorithms



Model-based

Explicit: Model

Model-Free:

No model

Find a good policy: Problem settings

Model-based control

(Agent's internal computation)

- Given model of how the world works
- Transition and reward models
- Algorithm computes how to act in order to maximize expected reward

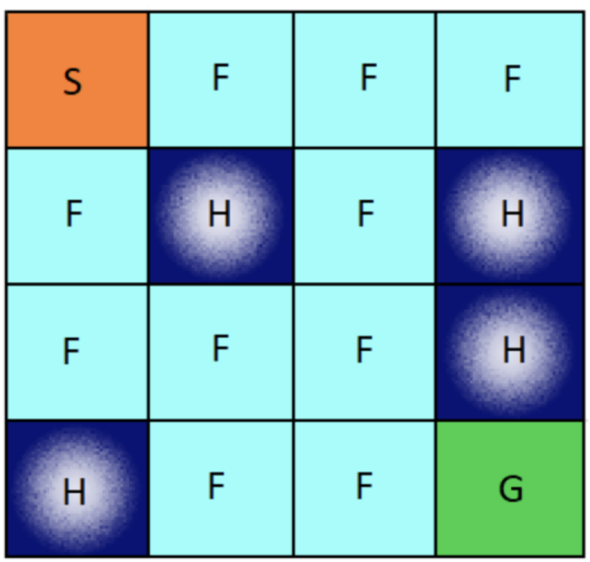
Model-free control

- Computing while interacting with environment
  - Agent doesn't know how world works
  - Interacts with world to implicitly/explicitly learn how world works
  - Agent improves policy (may involve planning)

Find a good policy: Problem settings

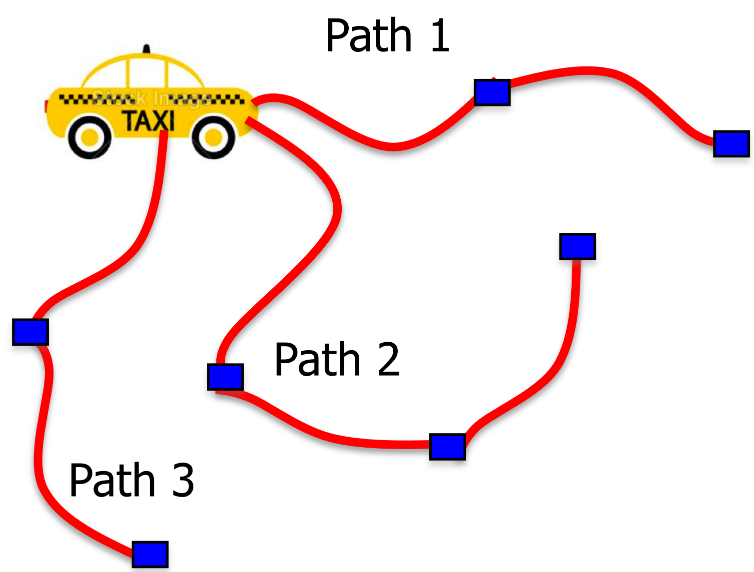
Model-based control

- (Agent's internal computation)
  - Frozen Lake project 1
  - Know all rules of game / perfect model
  - dynamic programming, tree search



Model-free control

- Computing while interacting with environment
  - Taxi passenger-seeking problem
  - Demand/Traffic dynamics are uncertain
  - Huge state space



Find a good policy: Problem settings

Model-based control

Given: MDP

- <S, A, P, R, γ>

Output:

- π

Model-free control

- Given: MDP without R, P
  - S, A, γ
- Unknow
  - P, R,
- Output:
  - π

This lecture

- Review:
  - Policy Iteration, and Value iteration

- Model-Free Policy Evaluation
  - Monte Carlo policy evaluation
  - Temporal-difference (TD) policy evaluation

MDP Policies

- Policy specifies what action to take in each state
  - Can be deterministic or stochastic
- For generality, consider as a conditional distribution
  - Given a state, specifies a distribution over actions
- Policy: $\pi(a|s) = P(a_t = a|s_t = s)$

MDP Policy Evaluation, Iterative Algorithm

For deterministic policy:

- Initialize $V_0(s) = 0$ for all s
- For $k = 1$ until convergence
    - For all s in S

$$V_k^\pi(s) = r(s, \pi(s)) + \gamma \sum_{s' \in S} p(s'|s, \pi(s)) V_{k-1}^\pi(s')$$

- This is a **Bellman backup** for a particular policy

MDP Policy Evaluation, Iterative Algorithm

For deterministic and stochastic policy:

Iterative Policy Evaluation, for estimating $V \approx v_\pi$

Input π, the policy to be evaluated
Algorithm parameter: a small threshold $\theta > 0$ determining accuracy of estimation
Initialize $V(s)$, for all $s \in \mathcal{S}^+$, arbitrarily except that $V(terminal) = 0$

Loop:
  $\Delta \leftarrow 0$
  Loop for each $s \in \mathcal{S}$:
    $v \leftarrow V(s)$
    $V(s) \leftarrow \sum_a \pi(a|s) \sum_{s',r} p(s', r \,|\, s, a)\big[r + \gamma V(s')\big]$
    $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
until $\Delta < \theta$

From: Reinforcement Learning: An Introduction, Sutton and Barto, 2nd Edition

MDP Control

- Compute the optimal policy

$$\pi^*(s) = \arg\max_{\pi} V^{\pi}(s)$$

- There **exists a unique optimal value function**
- Optimal policy for a MDP in an infinite horizon problem is deterministic

MDP Policy Iteration (PI)

- Set $i = 0$
- Initialize $\pi_0(s)$ randomly for all states s
- While $i == 0$ or $\|\pi_i - \pi_{i-1}\|_1 > 0$ (L1-norm, measures if the policy changed for any state):
  - $V^{\pi_i} \leftarrow$ MDP V function policy **evaluation** of π_i
  - $\pi_{i+1} \leftarrow$ Policy **improvement**
  - $i = i + 1$

Policy Improvement

- Compute state-action value of a policy π_i
  - For s in S and a in A:

$$Q^{\pi_i}(s,a) = R(s,a) + \gamma \sum_{s' \in S} P(s'|s,a) V^{\pi_i}(s')$$

- Compute new policy π_{i+1}, for all $s \in S$

$$\pi_{i+1}(s) = \arg\max_a Q^{\pi_i}(s,a) \ \forall s \in S$$

MDP Policy Iteration (PI) (All-in-one algorithm)

Policy Iteration (using iterative policy evaluation) for estimating $\pi \approx \pi_*$

1. Initialization
   $V(s) \in \mathbb{R}$ and $\pi(s) \in \mathcal{A}(s)$ arbitrarily for all $s \in \mathcal{S}$

2. Policy Evaluation
   Loop:
   $\quad \Delta \leftarrow 0$
   $\quad$ Loop for each $s \in \mathcal{S}$:
   $\quad\quad v \leftarrow V(s)$
   $\quad\quad V(s) \leftarrow \sum_{s',r} p(s',r \,|\, s,\pi(s)) \big[r + \gamma V(s')\big]$
   $\quad\quad \Delta \leftarrow \max(\Delta, |v - V(s)|)$
   until $\Delta < \theta$ (a small positive number determining the accuracy of estimation)

3. Policy Improvement
   policy-stable $\leftarrow$ *true*
   For each $s \in \mathcal{S}$:
   $\quad$ *old-action* $\leftarrow \pi(s)$
   $\quad \pi(s) \leftarrow \arg\max_a \sum_{s',r} p(s',r \,|\, s,a) \big[r + \gamma V(s')\big]$
   $\quad$ If *old-action* $\neq \pi(s)$, then *policy-stable* $\leftarrow$ *false*
   If *policy-stable*, then stop and return $V \approx v_*$ and $\pi \approx \pi_*$; else go to 2

From: Reinforcement Learning: An Introduction, Sutton and Barto, 2nd Edition

Value Iteration (VI) Deterministic policy

- Set $k = 1$
- Initialize $V_0(s) = 0$ for all states s
- Loop until [finite horizon, convergence]:
  - For each state s

$$V_{k+1}(s) = \max_a \left[R(s,a) + \gamma \sum_{s' \in S} P(s'|s,a) V_k(s') \right]$$

  - View as Bellman backup on value function

$$V_{k+1} = BV_k$$

$$\pi_{k+1}(s) = \arg\max_a \left[R(s,a) + \gamma \sum_{s' \in S} P(s'|s,a) V_k(s') \right]$$

Value Iteration (VI)

Value Iteration, for estimating $\pi \approx \pi_*$

Algorithm parameter: a small threshold $\theta > 0$ determining accuracy of estimation
Initialize $V(s)$, for all $s \in \mathcal{S}^+$, arbitrarily except that $V(terminal) = 0$

Loop:
| $\quad\Delta \leftarrow 0$
| $\quad$Loop for each $s \in \mathcal{S}$:
| $\qquad v \leftarrow V(s)$
| $\qquad V(s) \leftarrow \max_a \sum_{s',r} p(s',r|s,a)\big[r + \gamma V(s')\big]$
| $\qquad \Delta \leftarrow \max(\Delta, |v - V(s)|)$
until $\Delta < \theta$

Output a deterministic policy, $\pi \approx \pi_*$, such that
$\quad \pi(s) = \operatorname{argmax}_a \sum_{s',r} p(s',r|s,a)\big[r + \gamma V(s')\big]$

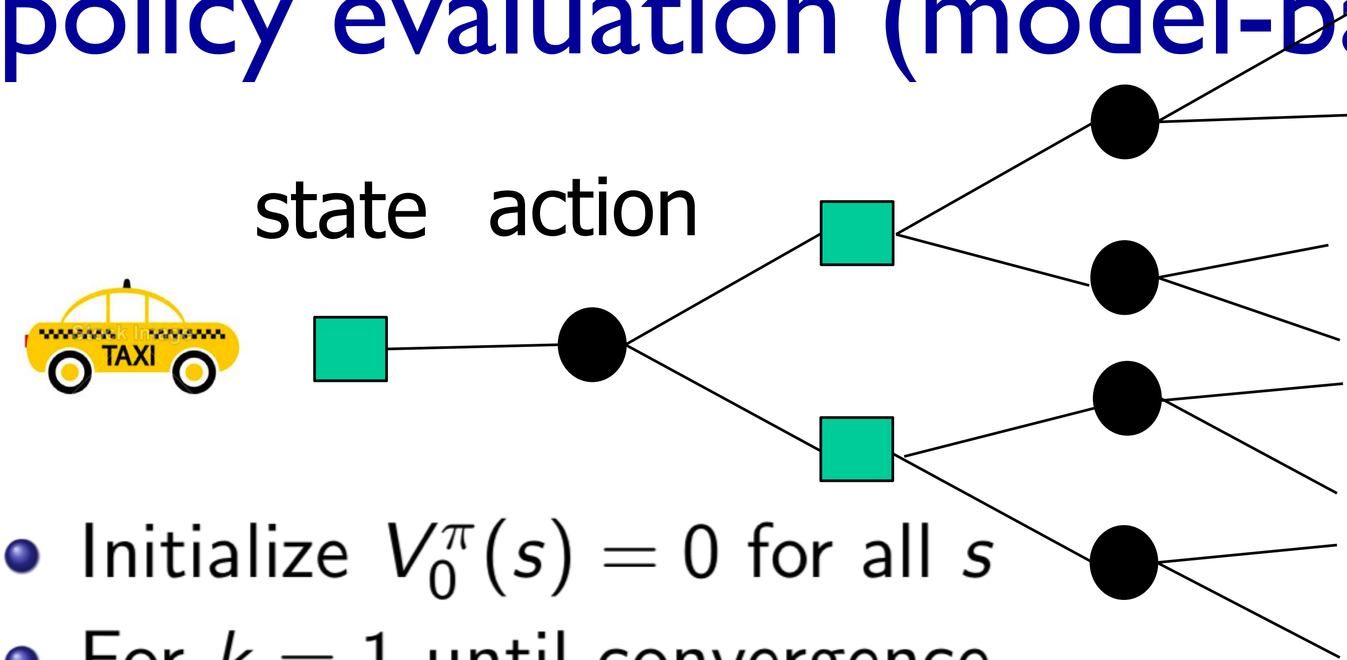
From: Reinforcement Learning: An Introduction, Sutton and Barto, 2nd Edition

This lecture

- Review:
  - Policy Iteration, and Value iteration

- Model-Free Policy Evaluation
  - Monte Carlo policy evaluation
  - Temporal-difference (TD) policy evaluation

Review of Dynamic Programming for policy evaluation (model-based)



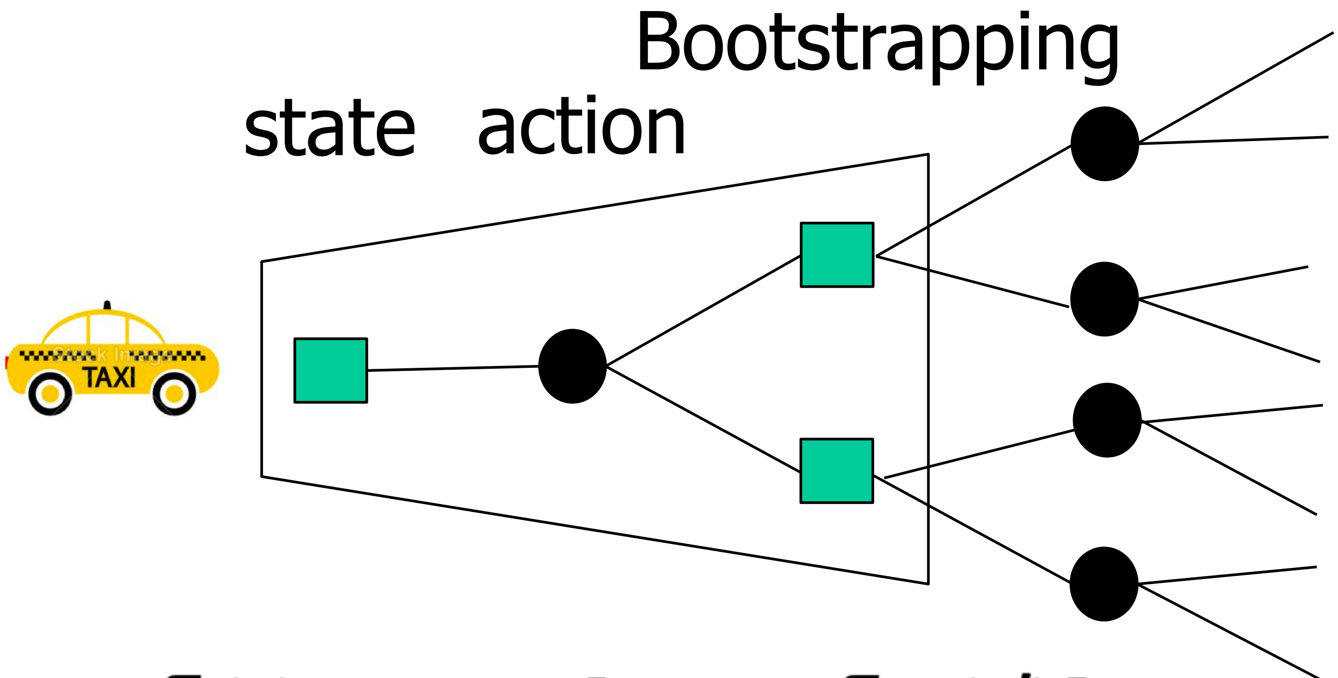
- Initialize $V_0^\pi(s) = 0$ for all s
- For $k = 1$ until convergence
  - For all s in S

$$V_k^\pi(s) = r(s, \pi(s)) + \gamma \sum_{s' \in S} p(s'|s, \pi(s)) V_{k-1}^\pi(s')$$

equivalently, $$V_k^\pi(s) = \sum_{a \in A} \sum_{s' \in S} \pi(a|s) P(s'|s,a)(r + \gamma V_{k-1}^\pi(s'))$$

$$V_k^\pi(s) = \mathbb{E}_{\pi,P}\left[r + \gamma V_{k-1}^\pi(s')\right]$$

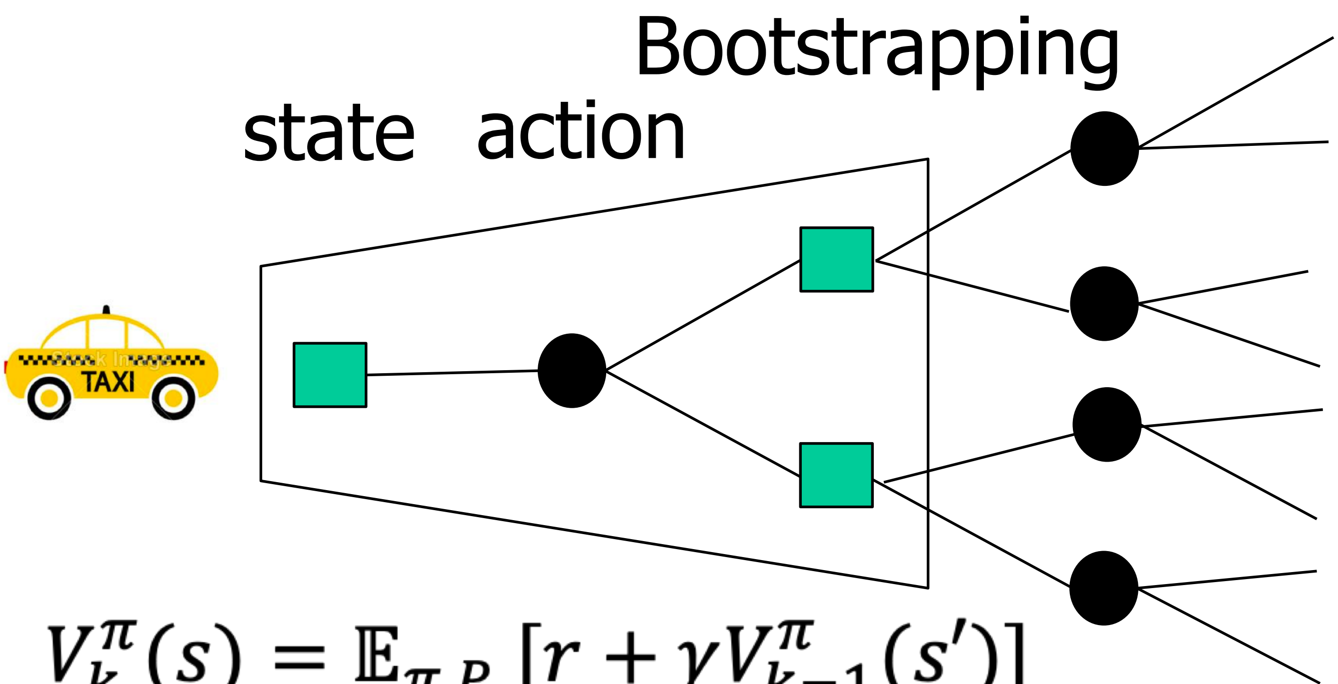
Review of Dynamic Programming for policy evaluation (model-based)



$$V_k^{\pi}(s) = \mathbb{E}_{\pi,P}\left[r + \gamma V_{k-1}^{\pi}(s')\right]$$

- Bootstrapping: Update for V uses an estimate
- Known model *P(s'|s,a)* and *r(s,a)*

Review of Dynamic Programming for policy evaluation (model-based)



$$V_k^{\pi}(s) = \mathbb{E}_{\pi,P}\left[r + \gamma V_{k-1}^{\pi}(s')\right]$$

- Requires model of MDP *P(s'|s,a)* and *r(s,a)*
- Bootstraps future return using value estimate
- Requires Markov assumption: bootstrapping regardless of history

Model-free Policy Evaluation

- What if don't know transition model P nor the reward model R?

- Today: ***Policy evaluation without a model***

- Given data and/or ability to interact in the environment Efficiently compute a good estimate of a policy π

Model-free Policy Evaluation

- Monte Carlo (MC) policy evaluation
  - First visit based
  - Every visit based

- Temporal Difference (TD)
  - TD(0)

- Metrics to evaluate and compare algorithms

Monte Carlo (MC) policy evaluation

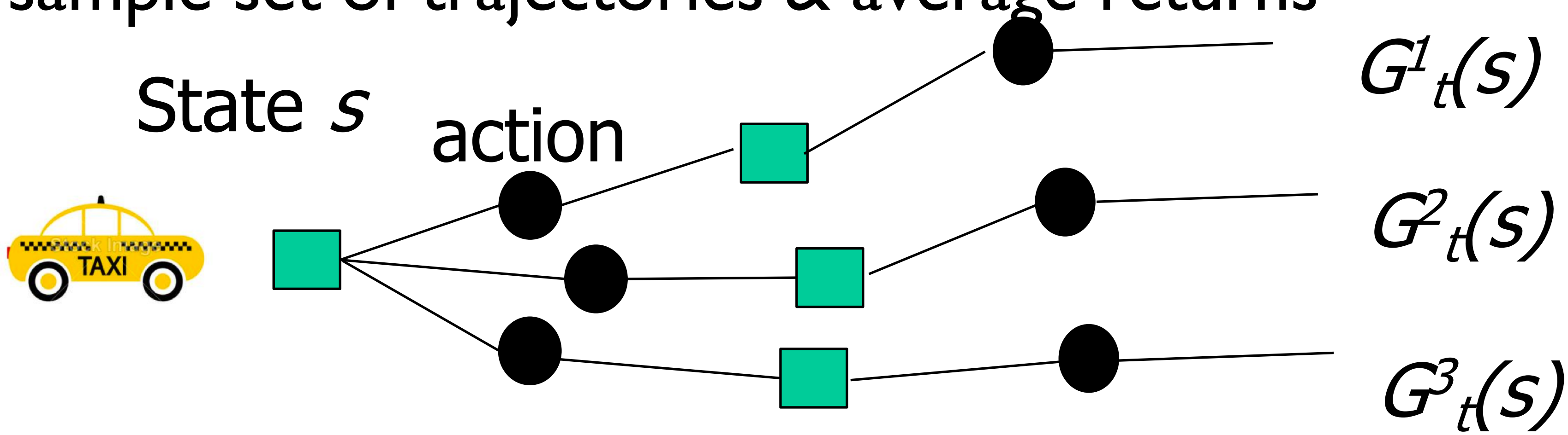
- Return of a trajectory under policy π

$$G_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \gamma^3 r_{t+3} + \cdots$$

- Value function:
  - Expectation over trajectories T generated by following π

$$V^\pi(s) = \mathbb{E}_{T\sim\pi}[G_t | s_t = s]$$

- Simple idea: Value = mean return
  - sample set of trajectories & average returns



Monte Carlo (MC) Policy Evaluation

- If trajectories are all finite, sample set of trajectories & average returns
- Does not require MDP dynamics/rewards
- No bootstrapping
- Does not assume state is Markov
- Can **only** be applied to episodic MDPs
    - Averaging over returns from a complete episode
    - Requires each episode to terminate

First-Visit Monte Carlo (MC) On Policy Evaluation

Initialize $N(s) = 0$, $G(s) = 0 \ \forall s \in S$

Loop

- Sample episode $i = s_{i,1}, a_{i,1}, r_{i,1}, s_{i,2}, a_{i,2}, r_{i,2}, \ldots, s_{i,T_i}$
- Define $G_{i,t} = r_{i,t} + \gamma r_{i,t+1} + \gamma^2 r_{i,t+2} + \cdots \gamma^{T_i - 1} r_{i,T_i}$ as return from time step t onwards in ith episode
- For each state s visited in episode i
  - For **first** time t that state s is visited in episode i
    - Increment counter of total first visits: $N(s) = N(s) + 1$
    - Increment total return $G(s) = G(s) + G_{i,t}$
    - Update estimate $V^{\pi}(s) = G(s)/N(s)$

First-Visit Monte Carlo (MC) On Policy Evaluation

Initialize $N(s) = 0$, $G(s) = 0$ $\forall s \in S$
Loop

- Sample episode $i = s_{i,1}, a_{i,1}, r_{i,1}, s_{i,2}, a_{i,2}, r_{i,2}, \ldots, s_{i,T_i}$
- Define $G_{i,t} = r_{i,t} + \gamma r_{i,t+1} + \gamma^2 r_{i,t+2} + \cdots \gamma^{T_i - 1} r_{i,T_i}$ as return from time step t onwards in ith episode
- For each state s visited in episode i
  - For $\boxed{\textbf{first} \text{ time } t}$ that state s is visited in episode i
    - Increment counter of total first visits: $N(s) = N(s) + 1$
    - Increment total return $G(s) = G(s) + G_{i,t}$
    - Update estimate $V^\pi(s) = G(s)/N(s)$

For example:

$s_1, a_1, r_1, s_2, a_2, r_2, s_2, a_3, r_3, \ldots \ldots$

First-Visit Monte Carlo (MC) On Policy Evaluation

Initialize $N(s) = 0,\ G(s) = 0\ \forall s \in S$

Loop

- Sample episode $i = s_{i,1}, a_{i,1}, r_{i,1}, s_{i,2}, a_{i,2}, r_{i,2}, \ldots, s_{i,T_i}$
- Define $G_{i,t} = r_{i,t} + \gamma r_{i,t+1} + \gamma^2 r_{i,t+2} + \cdots \gamma^{T_i - 1} r_{i,T_i}$ as return from time step t onwards in ith episode
- For each state s visited in episode i
  - For **first** time t that state s is visited in episode i
    - Increment counter of total first visits: $N(s) = N(s) + 1$
    - Increment total return $G(s) = G(s) + G_{i,t}$
    - Update estimate $V^\pi(s) = G(s)/N(s)$

Properties:

- V^π estimator is an unbiased estimator of true $\mathbb{E}_\pi[G_t | s_t = s]$
- By law of large numbers, as $N(s) \to \infty$, $V^\pi(s) \to \mathbb{E}_\pi[G_t | s_t = s]$

Model-free Policy Evaluation

- Monte Carlo (MC) policy evaluation
  - First visit based
  - Every visit based
- Temporal Difference (TD)
  - TD(0)
- Metrics to evaluate and compare algorithms

Every-Visit Monte Carlo (MC) On Policy Evaluation

Initialize $N(s) = 0,\ G(s) = 0\ \forall s \in S$
Loop

- Sample episode $i = s_{i,1}, a_{i,1}, r_{i,1}, s_{i,2}, a_{i,2}, r_{i,2}, \ldots, s_{i,T_i}$
- Define $G_{i,t} = r_{i,t} + \gamma r_{i,t+1} + \gamma^2 r_{i,t+2} + \cdots \gamma^{T_i-1} r_{i,T_i}$ as return from time step t onwards in ith episode
- For each state s visited in episode i
    - For **every** time t that state s is visited in episode i
        - Increment counter of total first visits: $N(s) = N(s) + 1$
        - Increment total return $G(s) = G(s) + G_{i,t}$
        - Update estimate $V^{\pi}(s) = G(s)/N(s)$

For example:

$s_1, a_1, r_1, s_2, a_2, r_2, s_2, a_3, r_3, \ldots \ldots$

Every-Visit Monte Carlo (MC) On Policy Evaluation

Initialize $N(s) = 0$, $G(s) = 0 \; \forall s \in S$

Loop

- Sample episode $i = s_{i,1}, a_{i,1}, r_{i,1}, s_{i,2}, a_{i,2}, r_{i,2}, \ldots, s_{i,T_i}$
- Define $G_{i,t} = r_{i,t} + \gamma r_{i,t+1} + \gamma^2 r_{i,t+2} + \cdots \gamma^{T_i - 1} r_{i,T_i}$ as return from time step t onwards in ith episode
- For each state s visited in episode i
  - For **every** time t that state s is visited in episode i
    - Increment counter of total first visits: $N(s) = N(s) + 1$
    - Increment total return $G(s) = G(s) + G_{i,t}$
    - Update estimate $V^\pi(s) = G(s)/N(s)$

Properties:

- V^π every-vist MC estimator is an **biased** estimator of V^π
- But consistent estimator and often has better MSE

$s_1, a_1, r_1, s_2, a_2, r_2, s_2, a_3, r_3, \ldots \ldots$

Incremental Monte Carlo (MC) On Policy Evaluation

After each episode $i = s_{i,1}, a_{i,1}, r_{i,1}, s_{i,2}, a_{i,2}, r_{i,2}, \ldots$

- Define $G_{i,t} = r_{i,t} + \gamma r_{i,t+1} + \gamma^2 r_{i,t+2} + \cdots$ as return from time step t onwards in ith episode
- For state s visited at time step t in episode i

Every visit

  - Increment counter of total first visits: $N(s) = N(s) + 1$
  - Update estimate

$$V^\pi(s) = V^\pi(s)\frac{N(s)-1}{N(s)} + \frac{G_{i,t}}{N(s)} = V^\pi(s) + \frac{1}{N(s)}(G_{i,t} - V^\pi(s))$$

- Increment total return $G(s) = G(s) + G_{i,t}$
- Update estimate $V^\pi(s) = G(s)/N(s)$

$s_1, a_1, r_1, s_2, a_2, r_2, s_2, a_3, r_3, \ldots \ldots$

Incremental Monte Carlo (MC) On Policy Evaluation, Running Mean

?

Initialize $N(s) = 0,\ G(s) = 0\ \forall s \in S$

Loop

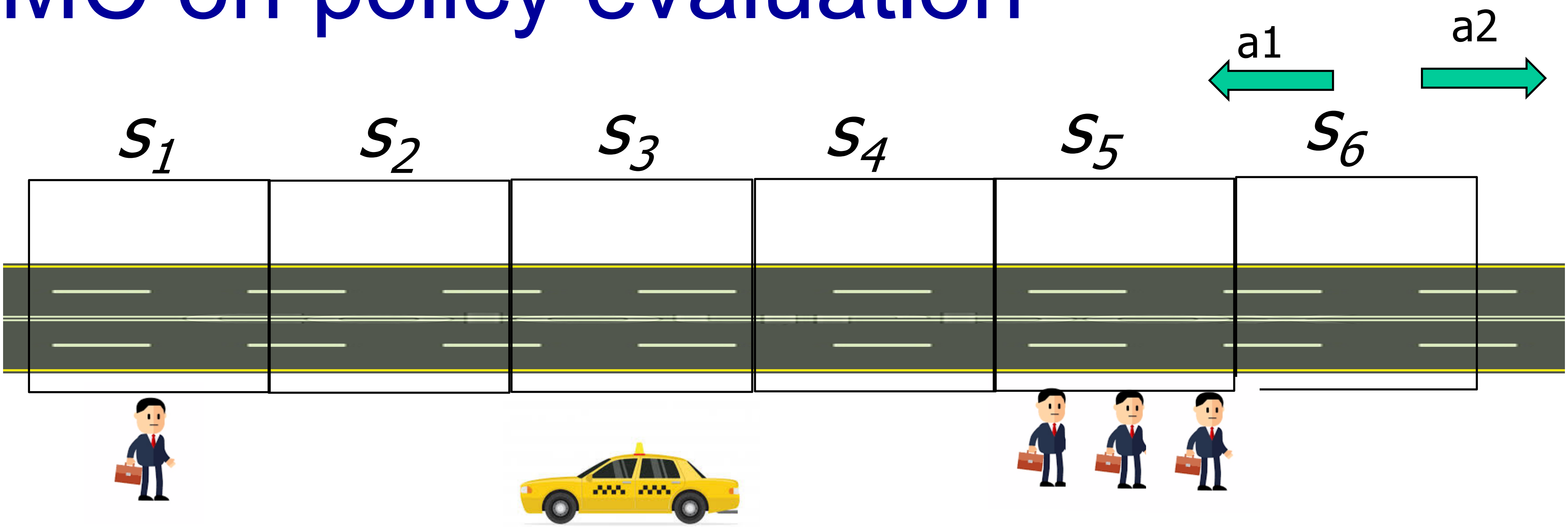
- Sample episode $i = s_{i,1}, a_{i,1}, r_{i,1}, s_{i,2}, a_{i,2}, r_{i,2}, \ldots, s_{i,T_i}$
- Define $G_{i,t} = r_{i,t} + \gamma r_{i,t+1} + \gamma^2 r_{i,t+2} + \cdots \gamma^{T_i - 1} r_{i,T_i}$ as return from time step t onwards in ith episode
- For state s visited at time step t in episode i
  - Increment counter of total first visits: $N(s) = N(s) + 1$
  - Update estimate

$$V^{\pi}(s) = V^{\pi}(s) + \alpha(G_{i,t} - V^{\pi}(s))$$

- $\alpha = \frac{1}{N(s)}$: identical to every visit MC
- $\alpha > \frac{1}{N(s)}$: forget older data, helpful for non-stationary domains

$s_1, a_1, r_1, s_2, a_2, r_2, s_2, a_3, r_3, \ldots \ldots$ How about $\alpha = 1$?

MC on policy evaluation



Taxi passenger-seeking process: R=[1,0,0,0,3,0]

For any action, $\pi(s) = a_1, \forall s, \gamma = 1$.

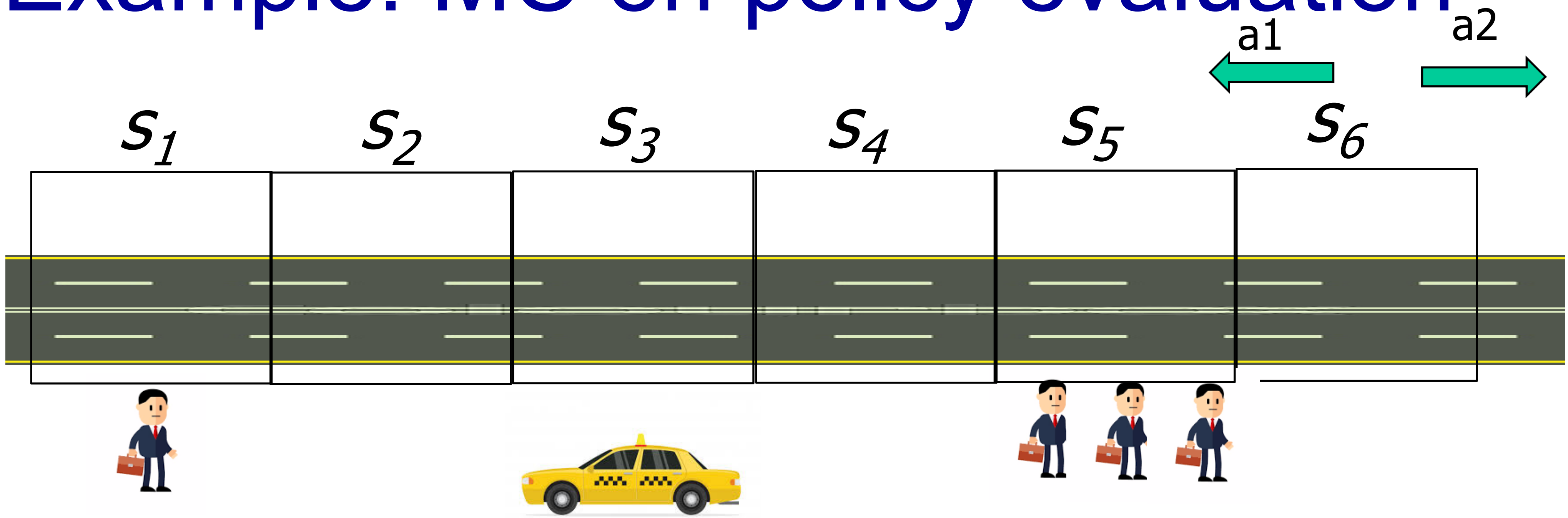
any action from s1 and s6 terminates episode

Given ($s_3, a_1, 0, s_3, a_1, 0, s_2, a_1, 0, s_1, a_1, 1, T$);

Q1: First visit MC estimate of V of each state?

Q2: Every visit MC estimate of s_2?

Example: MC on policy evaluation



Taxi passenger-seeking process: R=[1,0,0,0,3,0]

For any action, π(s) = a_1, ∀s, γ = 1.

any action from s1 and s6 terminates episode

Given (s_3, a_1, *0*, s_3, a_1, *0*, s_2, a_1, *0*, s_1, a_1, *1*, *T*);

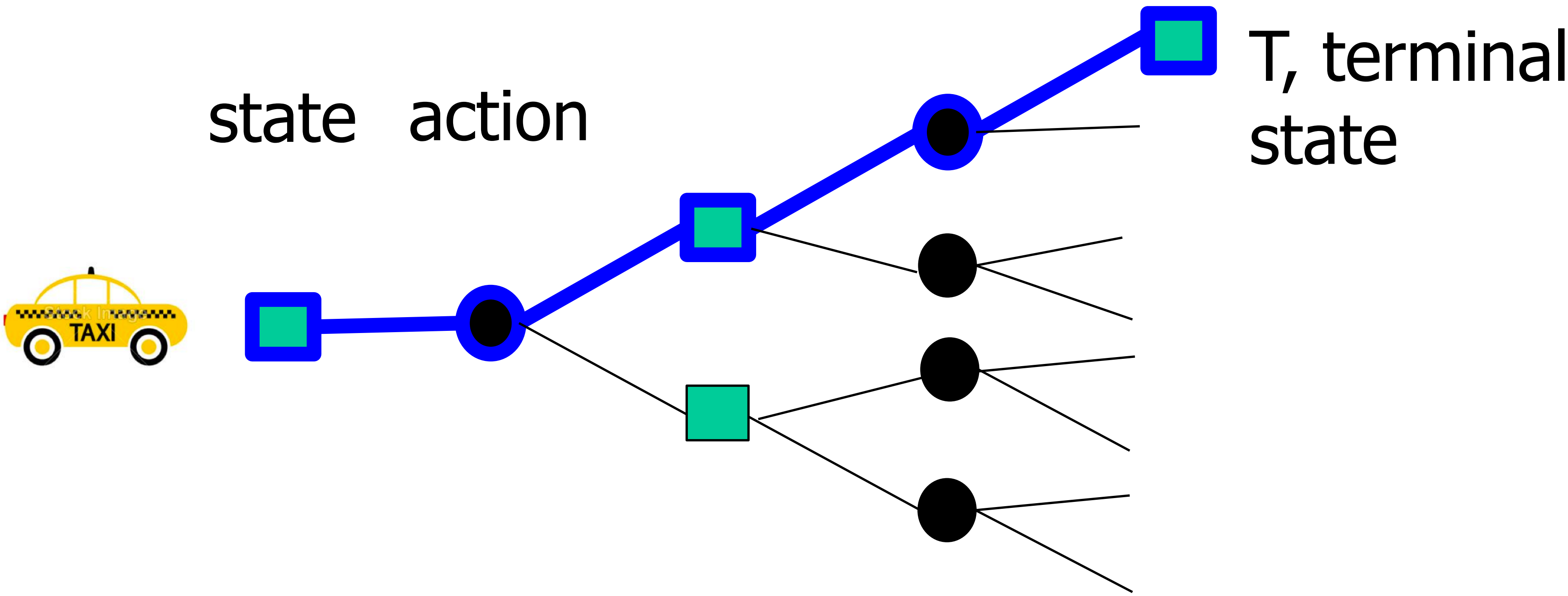
Q1: First visit MC estimate of V of each state?

V = [111000]

Q2: Every visit MC estimate of s_2? V(s_2)=1

MC policy evaluation

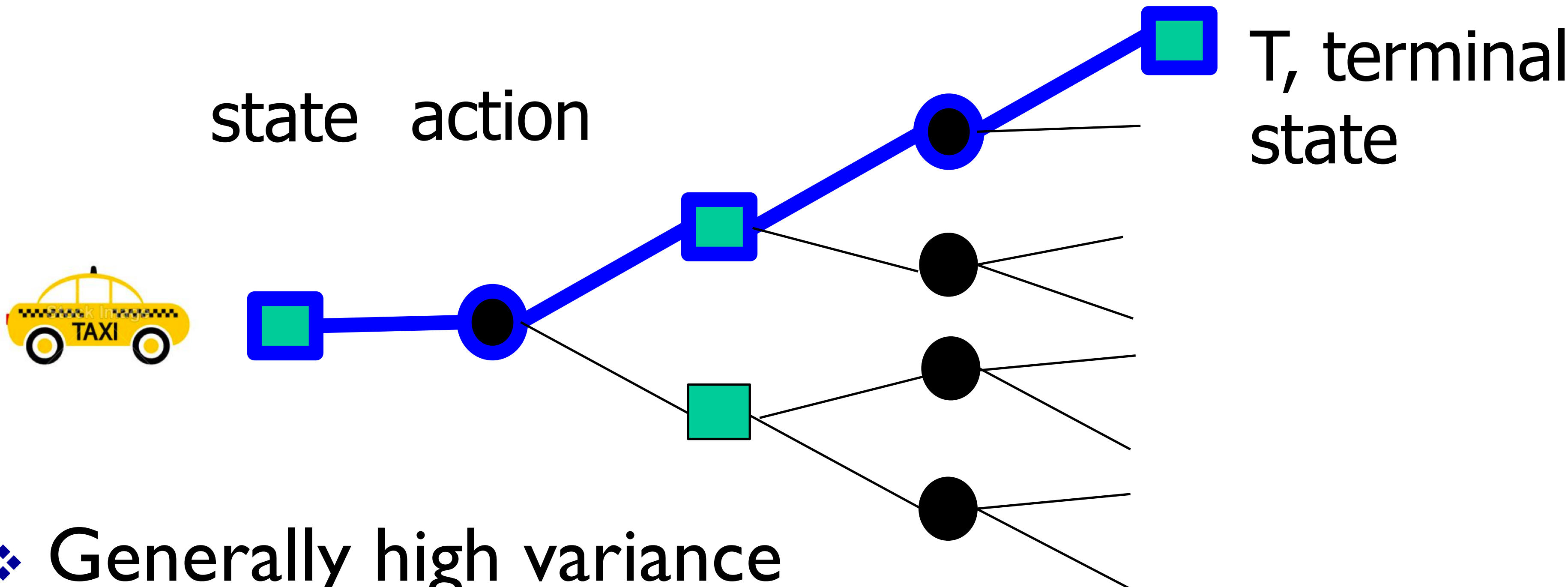
$$V^{\pi}(s) = V^{\pi}(s) + \alpha(G_{i,t} - V^{\pi}(s))$$



- MC updates the value estimate using a sample of the return to approximate an expectation

MC policy evaluation limitations

$$V^\pi(s) = V^\pi(s) + \alpha(G_{i,t} - V^\pi(s))$$



- Generally high variance
  - Reducing variance can require a lot of data
- Requires episodic settings
  - Episode must end before data from that episode can be used to update the value function

Model-free Policy Evaluation

- Monte Carlo (MC) policy evaluation
  - First visit based
  - Every visit based

- Temporal Difference (TD)
  - TD(0)
  - Combination of MC and Dynamic Programming

"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 2017

MC + DP = TD

DP is model based policy evaluation.

- Dynamic Programming (DP) policy evaluation

$$V_k^\pi(s) = \mathbb{E}_{\pi,P}\left[r + \gamma V_{k-1}^\pi(s')\right]$$

- Monte Carlo (MC) policy evaluation
  - $$V^\pi(s) = V^\pi(s) + \alpha(G_{i,t} - V^\pi(s))$$

- Temporal Difference (TD)

$$V^\pi(s) = V^\pi(s) + \alpha([r_t + \gamma V^\pi(s_{t+1})] - V^\pi(s))$$

Rewritten as

Temporal Difference [$TD(0)$] Learning

- Aim: estimate $V^\pi(s)$ given episodes generated under policy π
  - $s_1, a_1, r_1, s_2, a_2, r_2, \ldots$ where the actions are sampled from π
- Simplest TD learning: update value towards estimated value

$$V^\pi(s_t) = V^\pi(s_t) + \alpha(\underbrace{[r_t + \gamma V^\pi(s_{t+1})]}_{\text{TD target}} - V^\pi(s_t))$$

- TD error:

$$\delta_t = r_t + \gamma V^\pi(s_{t+1}) - V^\pi(s_t)$$

- Can immediately update value estimate after (s, a, r, s') tuple
- Don't need episodic setting

MC + DP = TD

DP is model based policy evaluation.

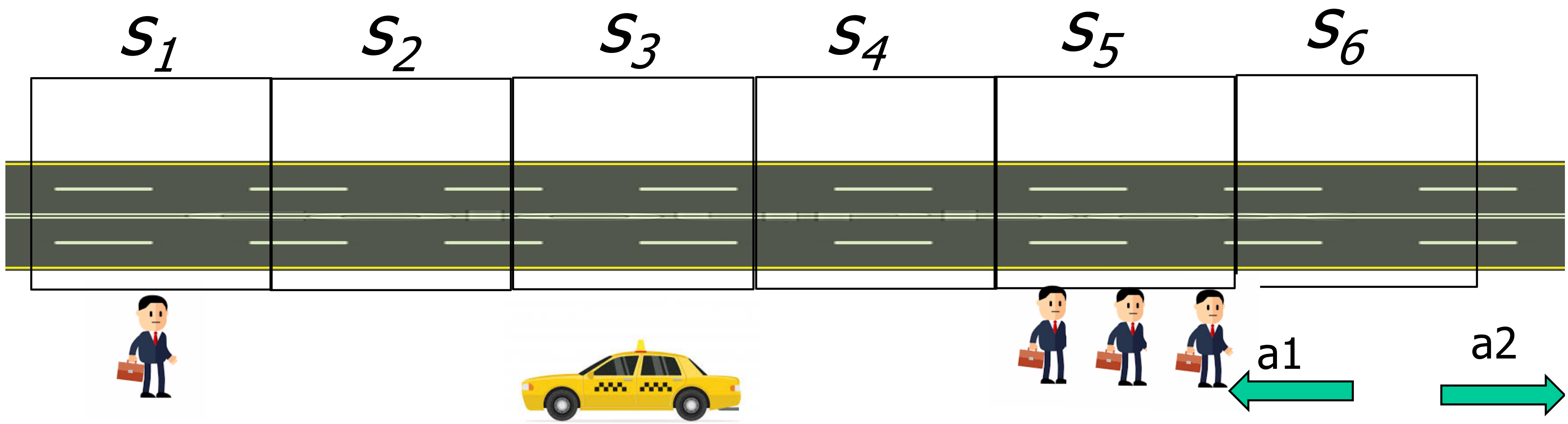
Input: α

Initialize $V^\pi(s) = 0, \forall s \in S$

Loop

- Sample **tuple** (s_t, a_t, r_t, s_{t+1})
- $V^\pi(s_t) = V^\pi(s_t) + \alpha(\underbrace{[r_t + \gamma V^\pi(s_{t+1})]}_{\text{TD target}} - V^\pi(s_t))$

Example: TD policy evaluation



Taxi passenger-seeking process: R=[1,0,0,0,3,0]

For any action, π(s) = $a_{1,}$ ∀s, γ = 1.

any action from s1 and s6 terminates episode

Given (s_3, a_1, 0, s_3, a_1, 0, s_2, a_1, 0, s_1, a_1, 1, T);

Q1: First visit MC estimate of V of each state?

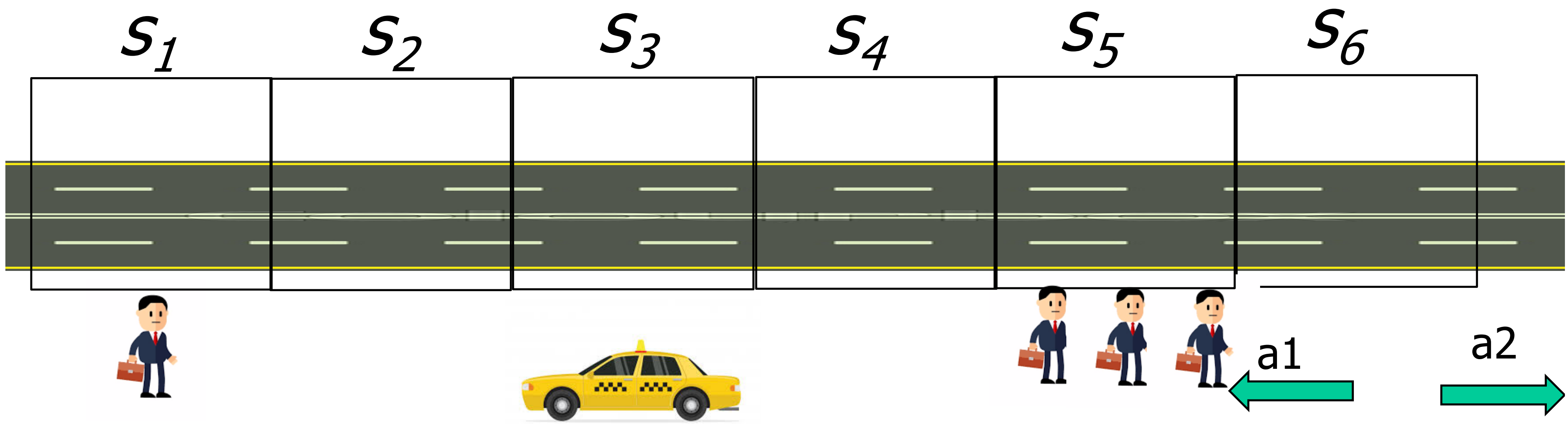
V = [111000]

$$V^{\pi}(s_t) = V^{\pi}(s_t) + \alpha(\underbrace{[r_t + \gamma V^{\pi}(s_{t+1})]}_{\text{TD target}} - V^{\pi}(s_t))$$

Q2: Every visit MC estimate of s_2? V(s_2)=1

Q3: TD estimate of all states (init at 0) with α = 1?

Example: TD policy evaluation



Taxi passenger-seeking process: R=[1,0,0,0,3,0]

For any action, $\pi(s) = a_1$, $\forall s$, $\gamma = 1$.

any action from s1 and s6 terminates episode

Given ($s_3, a_1, 0, s_3, a_1, 0, s_2, a_1, 0, s_1, a_1, 1, T$);

Q1: First visit MC estimate of V of each state?

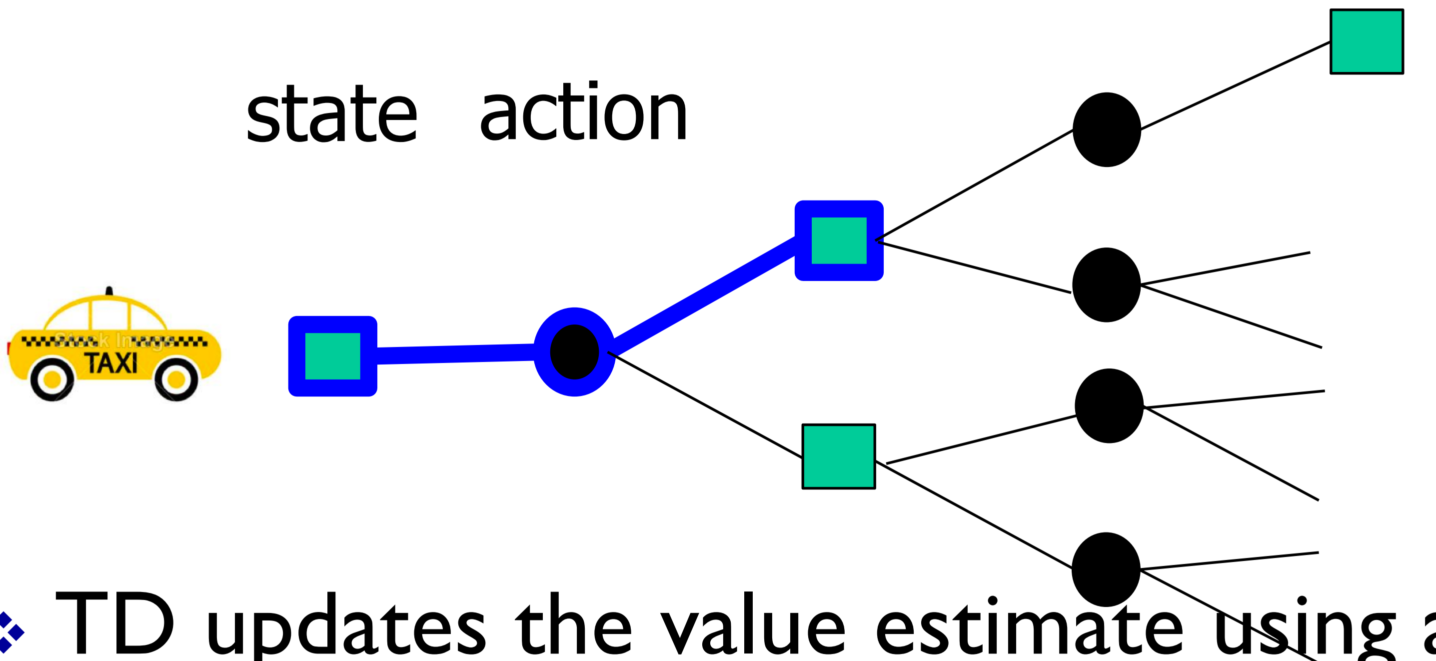
V = [111000] Q2: Every visit MC estimate of s_2? $V(s_2)=1$

Q3: TD estimate of all states (init at 0) with $\alpha = 1$?

V = [1 0 0 0 0 0 0]

TD(0) policy evaluation

$$V^\pi(s) = V^\pi(s) + \alpha([r_t + \gamma V^\pi(s_{t+1})] - V^\pi(s))$$



- TD updates the value estimate using a sample of s_{t+1} to approximate the expectation

- TD updates the value estimate by bootstrapping using estimate of $V(s_{t+1})$

Policy evaluation

	DP	MC	TD
Model-free method			
Handle non-episodic case			
Markovian assumption			

Policy evaluation

	DP	MC	TD
Model-free method	No	Yes	Yes
Handle non-episodic case	Yes	No	Yes
Markovian assumption	Yes	No	Yes

Next Lecture

- Review
  - Policy Iteration and Value Iteration

- Model-Free Policy Evaluation
  - Monte Carlo policy evaluation
  - Temporal-difference (TD) policy evaluation

- Model-Free Control
  - Monte Carlo control
  - Temporal-difference (TD) control
  - SARSA
  - Q-learning control

Quiz 2 Week #6 (2/23 W)

- Model-free Control
  - Model-free policy evaluation
    - Monte Carlo policy evaluation
    - TD policy evaluation
  - Model-free control
    - SARSA
    - Q-Learning
    - Double-Q-Learning

Any Comments & Critiques?

Bias, Variance, MSE

- Definition: the bias of an estimator $\hat{\theta}$ is:

$$Bias_\theta(\hat{\theta}) = \mathbb{E}_{x|\theta}[\hat{\theta}] - \theta$$

- Definition: the variance of an estimator $\hat{\theta}$ is:

$$Var(\hat{\theta}) = \mathbb{E}_{x|\theta}[(\hat{\theta} - \mathbb{E}[\hat{\theta}])^2]$$

- Definition: mean squared error (MSE) of an estimator $\hat{\theta}$ is:

$$MSE(\hat{\theta}) = Var(\hat{\theta}) + Bias_\theta(\hat{\theta})^2$$

- Biased vs unbiased estimator
  - Bias is zero or not,
- Consistent vs inconsistent estimator
  - When n goes to infinity, if the estimator goes to ground-truth

Bias/Variance of Model-free Policy Evaluation Algorithms

- Return G_t is an unbiased estimate of $V^\pi(s_t)$
- TD target $[r_t + \gamma V^\pi(s_{t+1})]$ is a biased estimate of $V^\pi(s_t)$
- But often much lower variance than a single return G_t
- Return function of multi-step sequence of random actions, states & rewards
- TD target only has one random action, reward and next state
- MC
  - Unbiased
  - High variance
  - Consistent (converges to true) even with function approximation
- TD
  - Some bias
  - Lower variance
  - TD(0) converges to true value with tabular representation

Convergence analysis

Policy Iteration

Delving Deeper Into Policy Improvement Step

$$Q^{\pi_i}(s,a) = R(s,a) + \gamma \sum_{s' \in S} P(s'|s,a) V^{\pi_i}(s')$$

Delving Deeper Into Policy Improvement Step

$$Q^{\pi_i}(s,a) = R(s,a) + \gamma \sum_{s' \in S} P(s'|s,a) V^{\pi_i}(s')$$

$$\max_a Q^{\pi_i}(s,a) \geq R(s,\pi_i(s)) + \gamma \sum_{s' \in S} P(s'|s,\pi_i(s)) V^{\pi_i}(s') = V^{\pi_i}(s)$$

$$\pi_{i+1}(s) = \arg\max_a Q^{\pi_i}(s,a)$$

- Suppose we take $\pi_{i+1}(s)$ for one action, then follow π_i forever
  - Our expected sum of rewards is at least as good as if we had always followed π_i
- But new proposed policy is to always follow π_{i+1} ...

Monotonic Improvement in Policy

- Definition
$$V^{\pi_1} \geq V^{\pi_2} : V^{\pi_1}(s) \geq V^{\pi_2}(s), \forall s \in S$$
- Proposition: $V^{\pi_{i+1}} \geq V^{\pi_i}$ with strict inequality if π_i is suboptimal, where π_{i+1} is the new policy we get from policy improvement on π_i

Proof: Monotonic Improvement in Policy

$$
\begin{aligned}
V^{\pi_i}(s) &\leq \max_a Q^{\pi_i}(s,a) \\
&= \max_a R(s,a) + \gamma \sum_{s' \in S} P(s'|s,a) V^{\pi_i}(s')
\end{aligned}
$$

Proof: Monotonic Improvement in Policy

$$
\begin{aligned}
V^{\pi_i}(s) \le & \max_a Q^{\pi_i}(s,a) \\
= & \max_a R(s,a) + \gamma \sum_{s' \in S} P(s'|s,a) V^{\pi_i}(s') \\
= & R(s, \pi_{i+1}(s)) + \gamma \sum_{s' \in S} P(s'|s, \pi_{i+1}(s)) V^{\pi_i}(s') \text{ //by the definition of } \pi_{i+1} \\
\le & R(s, \pi_{i+1}(s)) + \gamma \sum_{s' \in S} P(s'|s, \pi_{i+1}(s)) \left(\max_{a'} Q^{\pi_i}(s', a') \right) \\
= & R(s, \pi_{i+1}(s)) + \gamma \sum_{s' \in S} P(s'|s, \pi_{i+1}(s)) \\
& \left(R(s', \pi_{i+1}(s')) + \gamma \sum_{s'' \in S} P(s''|s', \pi_{i+1}(s')) V^{\pi_i}(s'') \right) \\
& \vdots \\
= & V^{\pi_{i+1}}(s)
\end{aligned}
$$

Convergence analysis

Value Iteration

Policy Iteration as Bellman Operations

- Bellman backup operator B^π for a particular policy is defined as

$$B^\pi V(s) = R^\pi(s) + \gamma \sum_{s' \in S} P^\pi(s'|s)V(s)$$

- Policy evaluation amounts to computing the fixed point of B^π
- To do policy evaluation, repeatedly apply operator until V stops changing

$$V^\pi = B^\pi B^\pi \cdots B^\pi V$$

Policy Iteration as Bellman Operations

- Bellman backup operator B^π for a particular policy is defined as

$$B^\pi V(s) = R^\pi(s) + \gamma \sum_{s' \in S} P^\pi(s'|s) V(s)$$

- To do policy improvement

$$\pi_{k+1}(s) = \arg\max_a R(s,a) + \gamma \sum_{s' \in S} P(s'|s,a) V^{\pi_k}(s')$$

Going Back to Value Iteration (VI)

- Set $k = 1$
- Initialize $V_0(s) = 0$ for all states s
- Loop until [finite horizon, convergence]:
  - For each state s

$$V_{k+1}(s) = \max_a R(s,a) + \gamma \sum_{s' \in S} P(s'|s,a) V_k(s')$$

  - Equivalently, in Bellman backup notation

$$V_{k+1} = BV_k$$

- To extract optimal policy if can act for $k+1$ more steps,

$$\pi(s) = \arg\max_a R(s,a) + \gamma \sum_{s' \in S} P(s'|s,a) V_{k+1}(s')$$

Contraction Operator

- Let O be an operator,and $|x|$ denote (any) norm of x
- If $|OV - OV'| \leq |V - V'|$, then O is a contraction operator

Will Value Iteration Converge?

- Yes, if discount factor $\gamma < 1$, or end up in a terminal state with probability 1
- Bellman backup is a contraction if discount factor, $\gamma < 1$
- If apply it to two different value functions, distance between value functions shrinks after applying Bellman equation to each

Proof: Bellman Backup is a Contraction on V for $\gamma < 1$

- Let $\|V - V'\| = \max_s |V(s) - V'(s)|$ be the infinity norm

$$\|BV_k - BV_j\| = \left\| \max_a \left(R(s,a) + \gamma \sum_{s' \in S} P(s'|s,a) V_k(s') \right) - \max_{a'} \left(R(s,a') + \gamma \sum_{s' \in S} P(s'|s,a') V_j(s') \right) \right\|$$

Proof: Bellman Backup is a Contraction on V for $\gamma < 1$

- Let $\|V - V'\| = \max_s |V(s) - V'(s)|$ be the infinity norm

$$
\begin{aligned}
\|BV_k - BV_j\| &= \left\| \max_a \left(R(s,a) + \gamma \sum_{s' \in S} P(s'|s,a) V_k(s') \right) - \max_{a'} \left(R(s,a') + \gamma \sum_{s' \in S} P(s'|s,a') V_j(s') \right) \right\| \\
&\leq \left\| \max_a \left(R(s,a) + \gamma \sum_{s' \in S} P(s'|s,a) V_k(s') - R(s,a) - \gamma \sum_{s' \in S} P(s'|s,a) V_j(s') \right) \right\| \\
&= \left\| \max_a \gamma \sum_{s' \in S} P(s'|s,a)(V_k(s') - V_j(s')) \right\| \\
&\leq \left\| \max_a \gamma \sum_{s' \in S} P(s'|s,a) \|V_k - V_j\|) \right\| \\
&= \left\| \gamma \|V_k - V_j\| \max_a \sum_{s' \in S} P(s'|s,a)) \right\| \\
&= \gamma \|V_k - V_j\|
\end{aligned}
$$

- Note: Even if all inequalities are equalities, this is still a contraction if $\gamma < 1$