Reinforcement Learning

Computer Engineering Department
Sharif University of Technology

Mohammad Hossein Rohban, Ph.D.

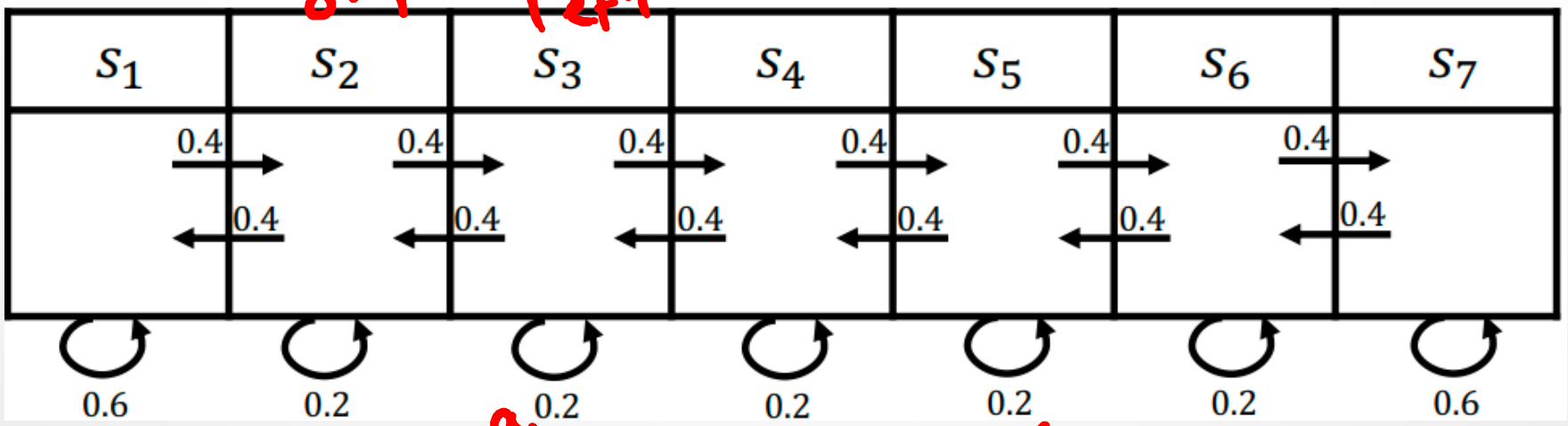
Spring 2024

Courtesy: Some slides are adopted from CS 285 Berkeley, and CS 234 Stanford, and Pieter Abbeel's compact series on RL.

Overview

- Last Lectures
  - Planning by dynamic programming to solve a known MDP
- This and next lectures:
  - Model-free **prediction** to estimate values in an unknown MDP
  - Model-free **control** to optimize values in an unknown MDP
  - Function approximation and (some) deep reinforcement learning
  - Off-policy learning

Example: Mars Rover



- Dynamics: $P(S_6 \mid S_5, a_2) = 0.4$, $P(S_2 \mid S_2, a_2) = 0.1$, …
- Reward: for all actions: +1 in state S_1 , +10 in state S_7 , 0 otherwise
- Let $\pi(s) = a_i$

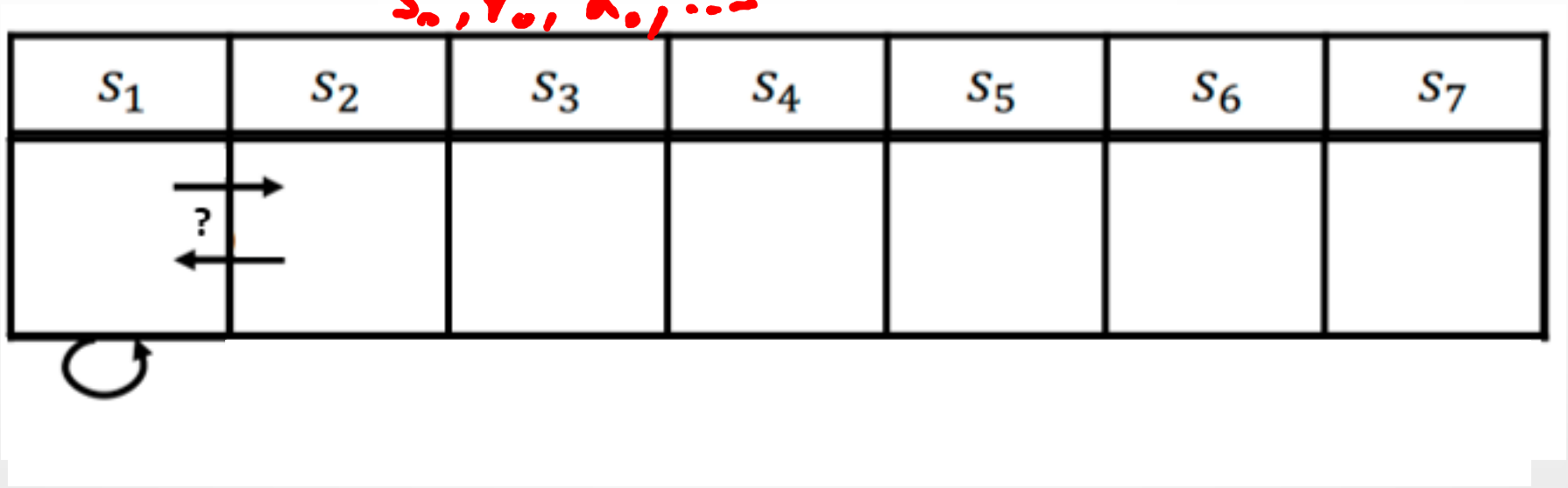
Monte-Carlo Prediction

Example: Mars Rover

trajectory

$S_0, a_0, S_1, a_1, S_2 \cdots$

$S_0, r_0, a_0, \cdots$



- Dynamics: $P(S_6 \mid S_5, a_1) = ?$, $P(S_2 \mid S_2, a_1) = ?$, …
- Reward: for all actions: +1 in state S_1 , +10 in state S_7 , 0 otherwise
- Let $\pi(s) = a_i$

Monte Carlo Methods - Introduction

$$V^{\pi}(s) := \mathbb{E}\left(R(s) + \gamma V^{\pi}(s') \mid s_0 = s \right)$$

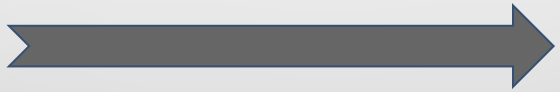
$$:= \mathbb{E}\left(\sum_{t=0}^{\infty} \gamma^t R_t \mid s_0 = s \right)$$

$$G_t = \sum_{s=t}^{\infty} \gamma^{s-t} R_s$$

- Experience samples to learn without a model
- MC methods require only experience— sample sequences of states, actions, and rewards from actual or simulated interaction with an environment.
- We can learn with samples: **episodes**!

We don't have access to

$$P(s'|s,a)$$

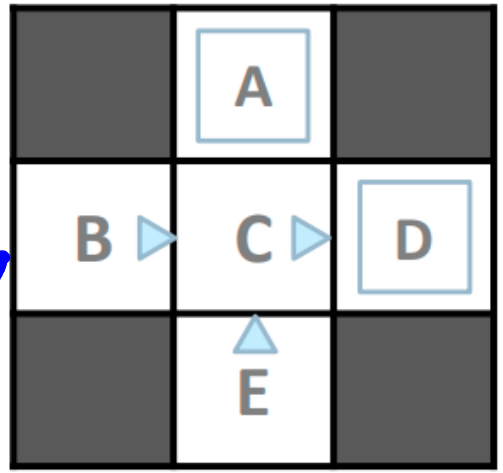


Model Free Learning!

Episodes: another example

episode i
C east D 1
D west C 1
C up A -10

Input Policy π



Assume: γ = 1

Observed Episodes (Training)

Episode 1

B, east, C, -1
C, east, D, -1
D, exit, x, +10

Episode 2

B, east, C, -1
C, east, D, -1
D, exit, x, +10

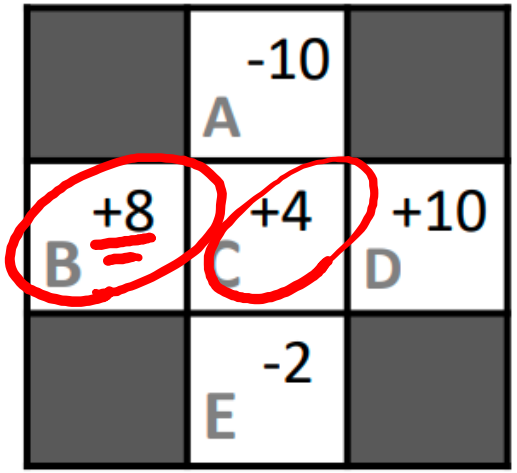
Episode 3

E, north, C, -1
C, east, D, -1
D, exit, x, +10

Episode 4

E, north, C, -1
C, east, A, -1
A, exit, x, -10

Output Values



Monte-Carlo prediction

- Suppose we wish to estimate $V_\pi(s)$, the value of a state s under policy π.
- The **first-visit mc** method estimates $V_\pi(s)$ as *the average of the returns* following first visits to s.

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 $Returns(S_t)$

$V(S_t) \leftarrow$ average$(Returns(S_t))$

terminal state

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

Example: Mars Rover

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

Loop

- Sample episode $i = s_{i,1}, a_{i,1}, r_{i,1}, s_{i,2}, a_{i,2}, r_{i,2}, \ldots, s_{i,T_i}$
- $G_{i,t} = r_{i,t} + \gamma r_{i,t+1} + \gamma^2 r_{i,t+2} + \cdots \gamma^{T_i - 1} r_{i,T_i}$
- For each time step t until T_i (the end of the episode i)
  - If this is the **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 $V^{\pi}(s) = G(s)/N(s)$
- Mars rover: R(s) = [1 0 0 0 0 0 +10]
- Trajectory = $(s_3, a_1, 0, s_2, a_1, 0, s_2, a_1, 0, s_1, a_1, 1, \text{terminal})$

$V^{\pi}(1) = 1$

$V^{\pi}(2) = 0.9$

$V^{\pi}(3) = 0.9^2$

$V^{\pi}(4) = 0$

$V^{\pi}(5) = 0$

$V^{\pi}(6) = 0$

$V^{\pi}(7) = 0$

$\gamma = 0.9$

$0 + \gamma 0 + \gamma^2 1$

What is V_{π} with First visit method?

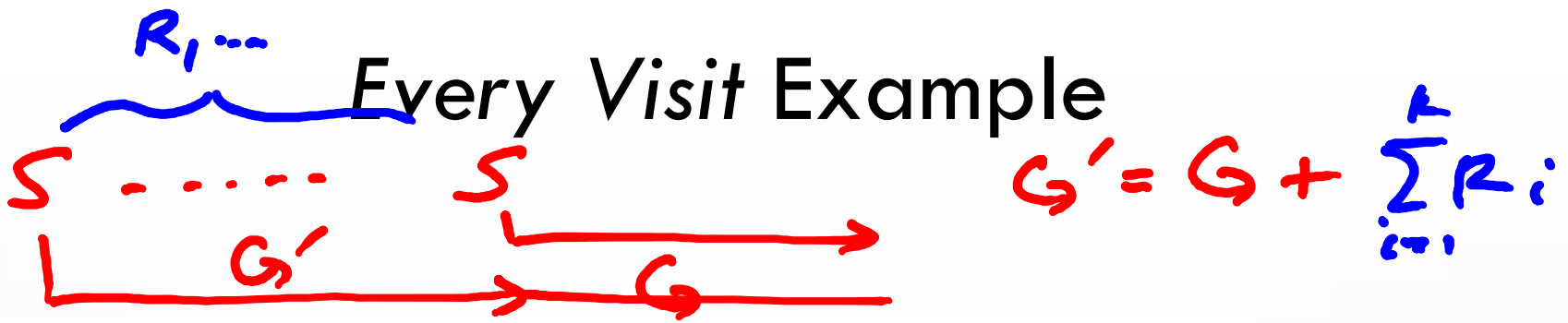
Every Visit Monte-Carlo Policy

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 time step t until T_i (the end of the episode i)
  - state s is the state visited at time step t in episodes i
  - Increment counter of total visits: $N(s) = N(s) + 1$
  - Increment total return $G(s) = G(s) + G_{i,t}$
  - Update estimate $V^{\pi}(s) = G(s)/N(s)$

Every Visit Example



- Mars rover: R(s) = [1 0 0 0 0 0 +10]
- Trajectory = (s_3, a_1, 0, s_2, a_1, 0, s_2, a_1, 0, s_1, a_1, 1, terminal)

Every visit method value of S2?

$\gamma = 0.9$

$V^{\pi}(1) = 1$

$V^{\pi}(2) = \frac{\gamma + \gamma^2}{2}$

$V^{\pi}(3) = \gamma^3$

$\vdots$

Incremental Monte-Carlo Policy

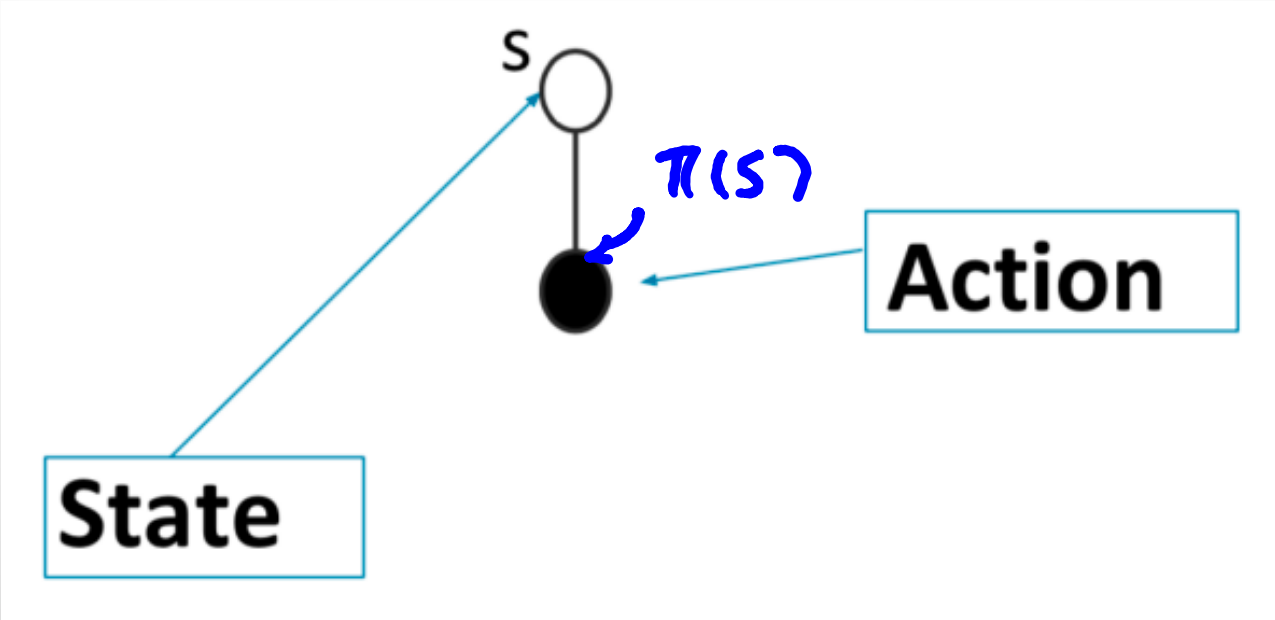
After each episode $i = s_{i,1}, a_{i,1}, r_{i,1}, s_{i,2}, a_{i,2}, r_{i,2}, \ldots$

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

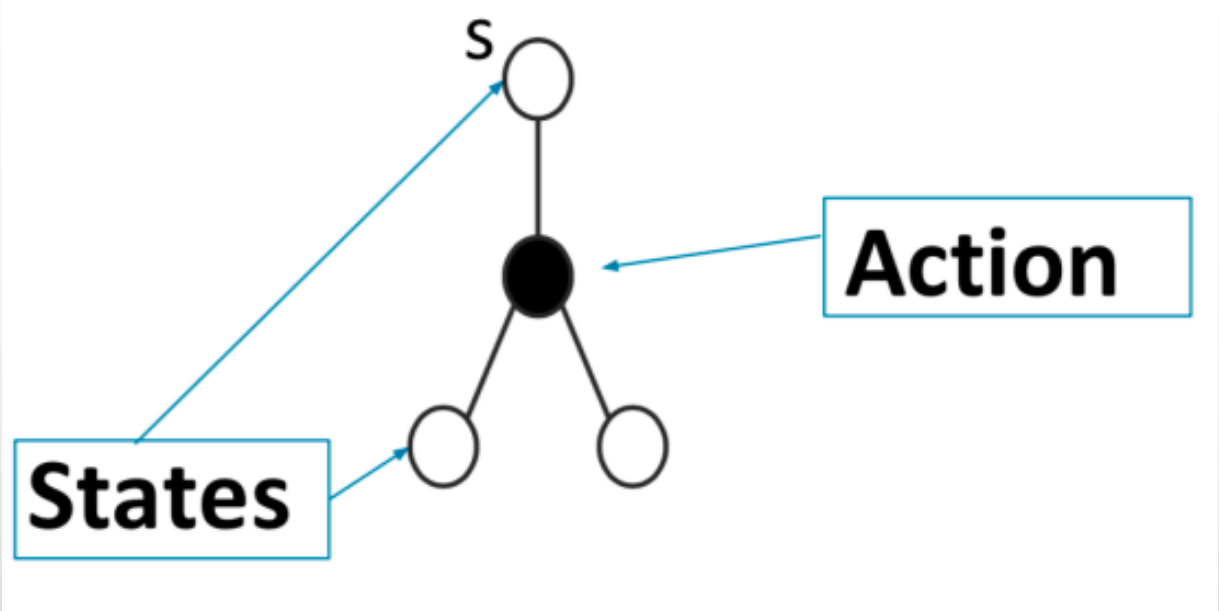
$$V^{\pi}(s) = V^{\pi}(s)\frac{N(s)-1}{N(s)} + \frac{G_{i,t}}{N(s)} = V^{\pi}(s) + \frac{1}{N(s)}(G_{i,t} - V^{\pi}(s))$$

$\alpha(s,i)$

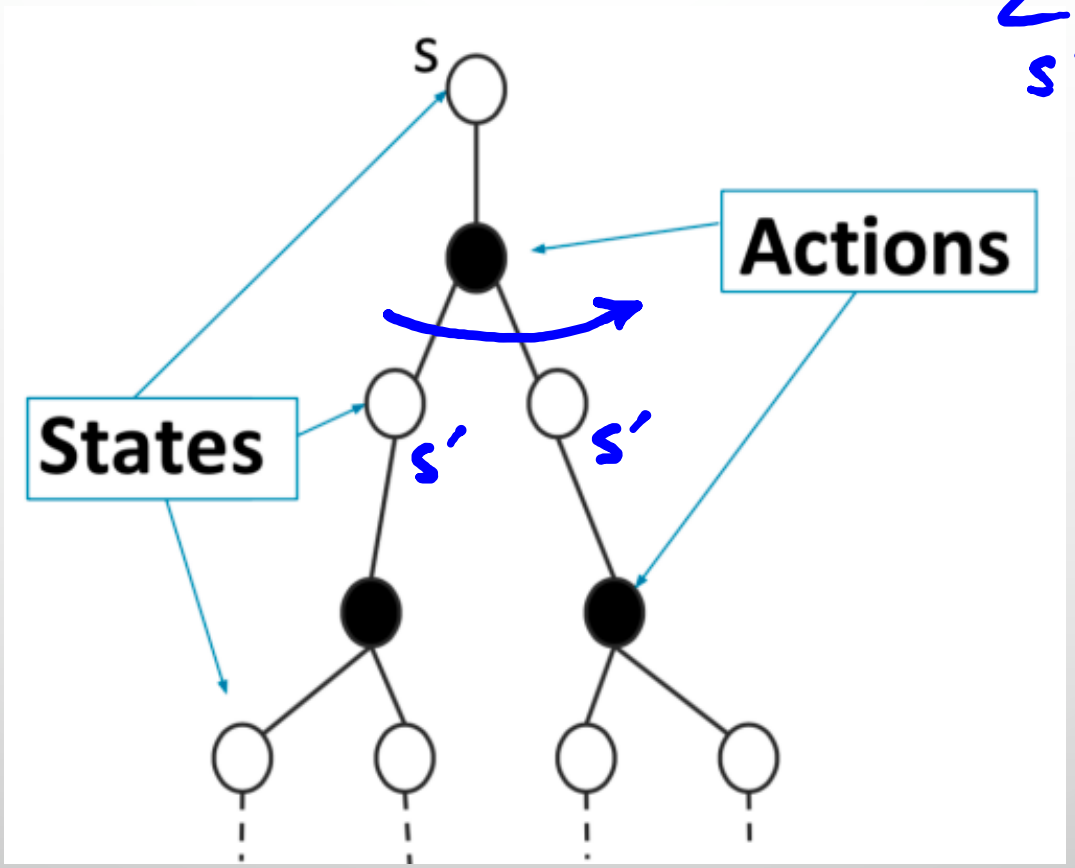
Policy Evaluation Diagram



Policy Evaluation Diagram



Policy Evaluation Diagram

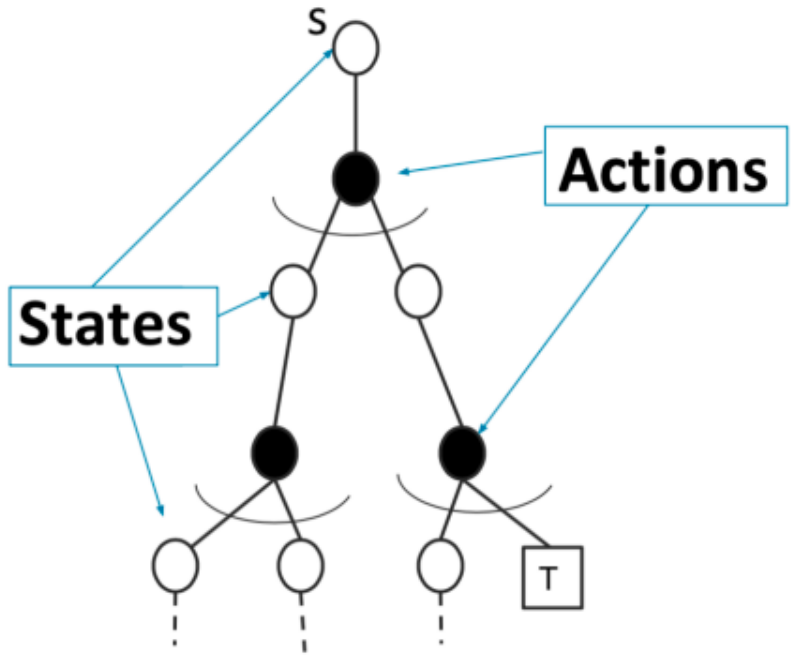


$$\sum_{s'} P(s'|s,a)[\dots]$$

$$R(s) + \gamma V^{\pi}(s')$$

Policy Evaluation Diagram

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



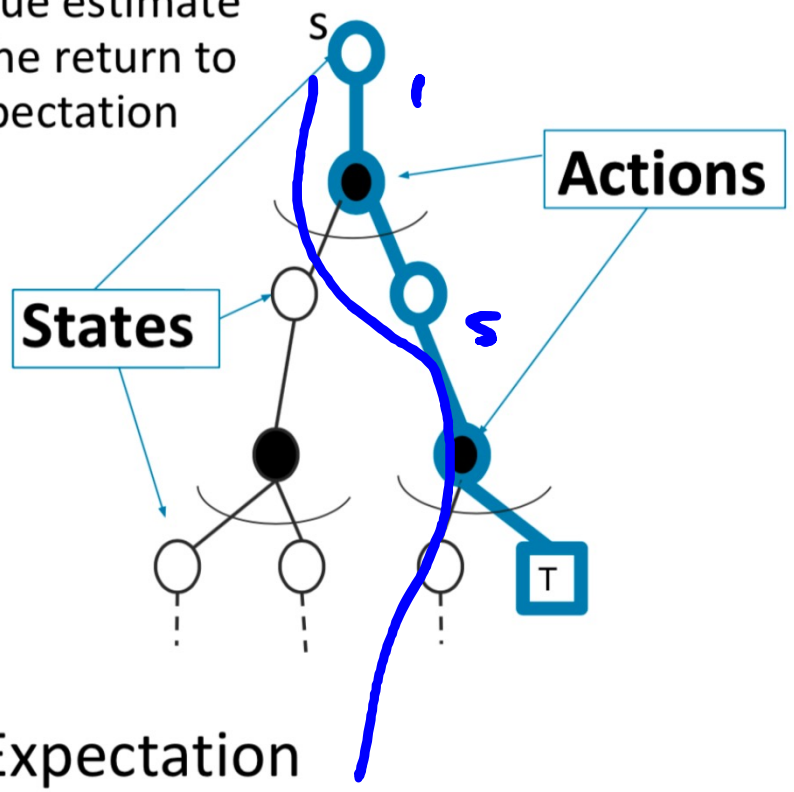
= Expectation

T = **Terminal state**

Policy Evaluation Diagram

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

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



Monte-Carlo as an Estimator

- Consider a statistical model that is parameterized by θ and that determines a probability distribution over observed data $P(x|\theta)$
- Consider a statistic $\hat{\theta}$ that provides an estimate of θ and is a function of observed data x
- Definition: the bias of an estimator $\hat{\theta}$ is:

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

- Let n be the number of data points x used to estimate the parameter θ and call the resulting estimate of θ using that data $\hat{\theta}_n$
- Then the estimator $\hat{\theta}_n$ is consistent if, for all $\epsilon > 0$

$$\lim_{n\to\infty} Pr(|\hat{\theta}_n - \theta| > \epsilon) = 0$$

- If an estimator is unbiased (bias $= 0$) is it consistent?

Properties of Monte-Carlo

$\hat{V}^{\pi}(s) \xrightarrow{P} \mathbb{E}_{\pi}(G_t | S_t = s)$

$N(s) \to \infty$

Properties:

- First-visit Monte Carlo
  - V^{π} estimator is an unbiased estimator of true $\mathbb{E}_{\pi}[G_t | s_t = s]$
  - By law of large numbers, as $N(s) \to \infty$, $V^{\pi}(s) \to \mathbb{E}_{\pi}[G_t | s_t = s]$
- Every-visit Monte Carlo
  - V^{π} every-visit MC estimator is a **biased** estimator of V^{π}
  - But consistent estimator and often has better MSE
- Incremental Monte Carlo
  - Properties depends on the learning rate α

$\alpha = \frac{1}{N(s)}$

$V^{\pi}(s) \leftarrow V^{\pi}(s) + \alpha_{t(s)}(G_t - V^{\pi}(s))$

Properties of Monte-Carlo

$$V_{n+1}^{\pi}(s_j) = V_n^{\pi}(s_j) + \alpha_n\left(G_i - V_n^{\pi}(s_j)\right) = (1-\alpha_n) V_n^{\pi}(s_j) + \alpha_n G_i$$

$$= (1-\alpha_n)(1-\alpha_{n-1}) V_{n-1}^{\pi}(s_j) + \alpha_{n-1}(1-\alpha_n) G_i^{n-1} + \alpha_n G_i^{n}$$

- Update is: $V^{\pi}(s_{it}) = V^{\pi}(s_{it}) + \alpha_k(s_j)(G_{i,t} - V^{\pi}(s_{it}))$
- where we have allowed α to vary (let k be the total number of updates done so far, for state $s_{it} = s_j$)
- If

$$\begin{cases} \sum_{n=1}^{\infty} \alpha_n(s_j) = \infty, \\ \sum_{n=1}^{\infty} \alpha_n^2(s_j) < \infty \end{cases}$$

$\alpha_n = \frac{1}{n}$ $\quad \frac{1}{n^{0.66}}$ $\quad \alpha_n = \frac{1}{\sqrt{n}}$

- then incremental MC estimate will converge to true value of the policy $V^{\pi}(s_j)$

First vs. Every visit MC

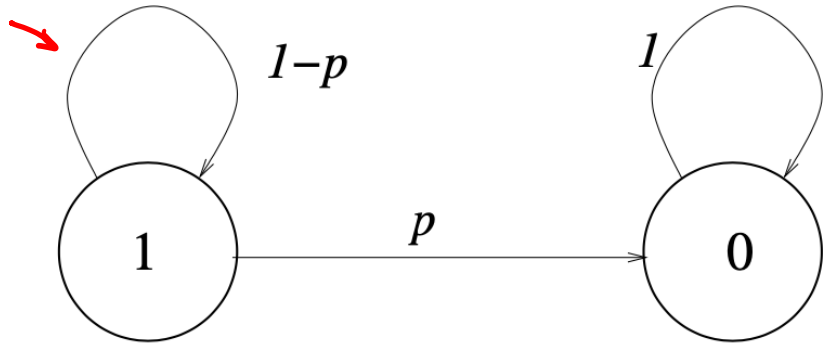
$V(0) = 0$

$V(1) = (1-p)[1 + V(1)] + p[1 + V(0)]$

$= 1 + (1-p)V(1)$

$\Rightarrow V(1) = 1/p$

Example: 2-state Markov Chain



The reward is 1 while in state 1 (while is 0 in the terminal state). All trajectories are $(x_0 = 1, x_1 = 1, \ldots, x_T = 0)$. By Bellman equations

$$V(1) = 1 + (1-p)V(1) + 0 \cdot p = \frac{1}{p},$$

First vs. Every visit MC

$\hat{V}(1) = \frac{T_1 + T_2 + \cdots + T_N}{N} \Rightarrow \mathbb{E}(\hat{V}(1)) = \frac{N \cdot \mathbb{E}(T_i)}{N}$

First-visit Monte-Carlo. All the trajectories start from state 1, then the return over one single trajectory is exactly T, i.e., $\widehat{V} = T$. $= 1/p$

The time-to-end T is a *geometric* r.v. with expectation

$$\mathbb{E}[\widehat{V}] = \mathbb{E}[T] = \frac{1}{p} = V^{\pi}(1) \Rightarrow \textit{unbiased estimator.}$$

Thus the MSE of $\widehat{V}$ coincides with the variance of T, which is

$N = 1$

$$\mathbb{E}\left[\left(T - \frac{1}{p}\right)^2\right] = \frac{1}{p^2} - \frac{1}{p}.$$

$MSE = 0 + \frac{1}{p^2} - \frac{1}{p}$

$= 100 - 10 = 90$

First vs. Every visit MC

episode 1

$\hat{V}_{(N)}(1) = \hat{V}_1(1) + \cdots + \hat{V}_N(1)$

$\mathbb{E}(\hat{V}(1)) = \mathbb{E}\left(\frac{T+1}{2}\right) = \frac{1/p+1}{2}$

Every-visit Monte-Carlo. Given one trajectory, we can construct $T-1$ sub-trajectories (number of times state 1 is visited), where the t-th trajectory has a return $T-t$.

$\mathrm{Var}\left[\hat{V}(1)\right]$

$$\hat{V} = \frac{1}{T}\sum_{t=0}^{T-1}(T-t) = \frac{1}{T}\sum_{t'=1}^{T} t' = \frac{T+1}{2}.$$

Bias $= 1/p - \frac{1/p+1}{2} = 1/2p - 1/2$

The corresponding expectation is

$$\mathbb{E}\left[\frac{T+1}{2}\right] = \frac{1+p}{2p} \neq V^{\pi}(1) \Rightarrow \textit{biased estimator}.$$

$\mathrm{Var}(\hat{V}) = \mathrm{Var}\left(\frac{T+1}{2}\right) = 1/4\left(\frac{1}{p^2} - \frac{1}{p}\right)$

$\mathrm{MSE} = \mathrm{Bias}^2 + \mathrm{Var} = (4.5)^2 + 22.5$

Disadvantages of Monte-Carlo Learning

$$\pi(s) = \begin{cases} \max_a \hat{Q}^{\pi}(s,a) & 1-\epsilon \\ \text{Random} & \epsilon \end{cases}$$

- We have seen MC algorithms can be used to learn value predictions

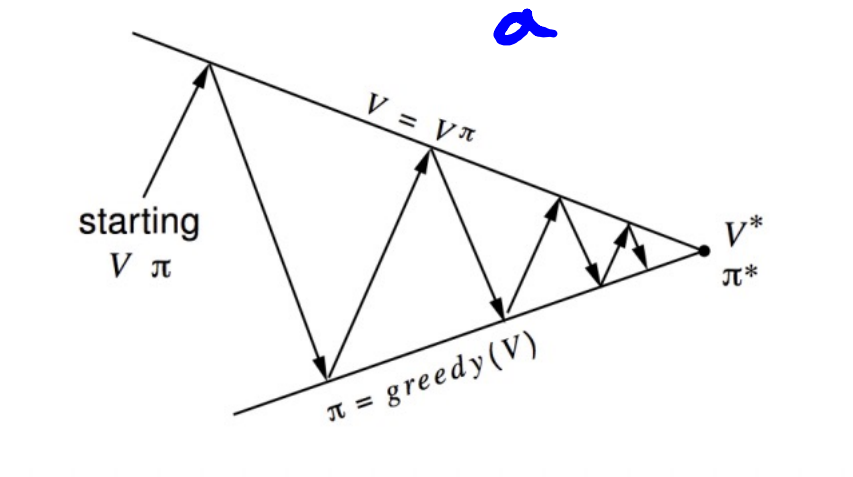
- But when episodes are long, learning can be slow
  - we have to **wait until an episode ends** before we can learn…
  - return can have **high variance** → ? $MSE = Bias^2 + Var$
    - Which one is more? First-visit or every-visit

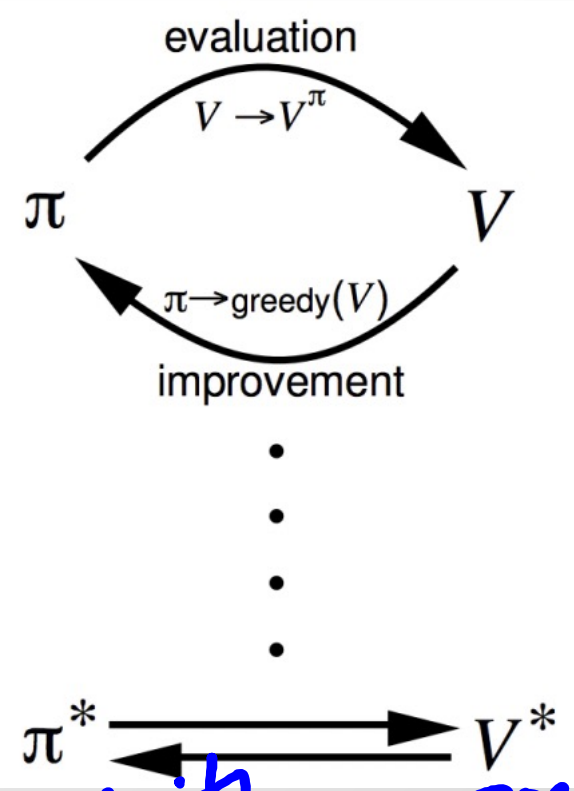
- Are there alternatives? (Spoiler: yes)

Monte-Carlo Control

Generalized Policy Iteration (Refresher)

$\pi'(s) = \arg\max_a Q^{\pi}(s,a)$ with prob. $1-\epsilon$

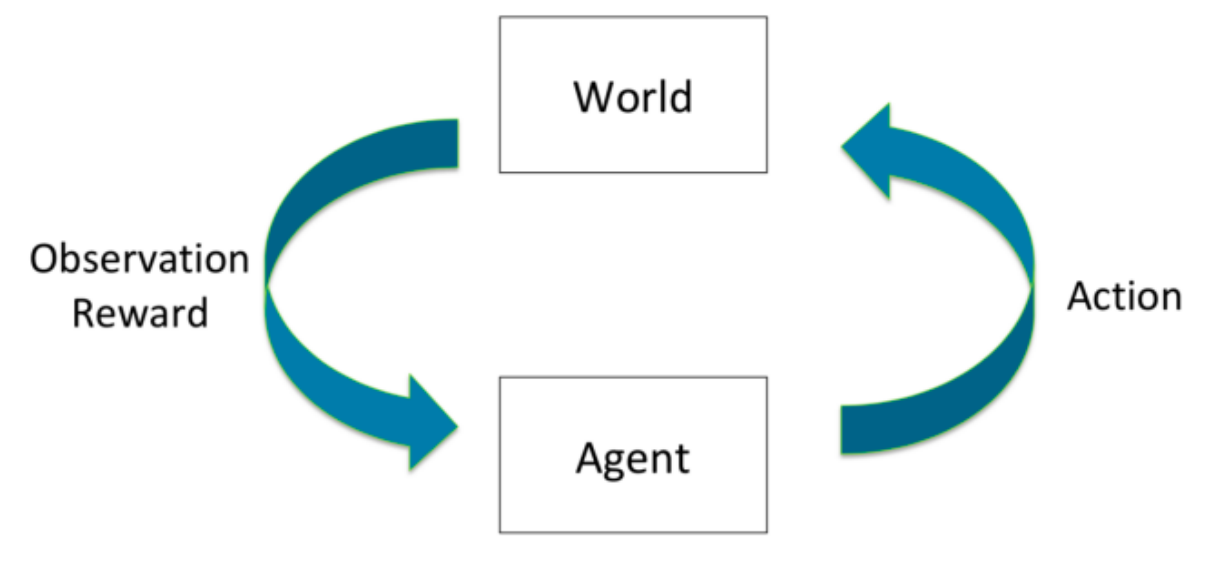




- Policy evaluation: Estimate $v_\pi(s)$ for all s
- Policy improvement: Generate π'such $v_{\pi'}(s) \geq v_\pi(s)$ for all s

$\pi'(s)$ = random action with prob. ϵ

Exploration Problem



- Goal: Learn to select actions to maximize total expected future reward
- Problem: Can't learn about actions without trying them (need to explore)
- Problem: But if we try new actions, spending less time taking actions that our past experience suggests will yield high reward (need to exploit knowledge of domain to achieve high reward)

Exploration vs Exploitation

Example

- Let's assume the following trajectory is sampled.
- $[s_1, a_1, 0, s_2, a_2, 0, s_3, a_2, 10]$.
- What actions would be selected in the improved policy, once the Q-values are updated?

Epsilon Greedy Policy

- Simple idea to balance exploration and achieving rewards
- Let $|A|$ be the number of actions
- Then an ϵ-greedy policy w.r.t a state action value $Q(s,a)$ is

$$\pi(a|s) =$$

- $\arg\max_a Q(s,a)$, w. prob $1-\epsilon+\frac{\epsilon}{|A|}$
- $a' \neq \arg\max Q(s,a)$ w. prob $\frac{\epsilon}{|A|}$

Theorem

For any ϵ-greedy policy π_i, the ϵ-greedy policy w.r.t. Q^{π_i}, π_{i+1} is a monotonic improvement $V^{\pi_{i+1}} \geq V^{\pi_i}$

$$Q^{\pi_i}(s, \pi_{i+1}(s)) = \sum_{a \in A} \pi_{i+1}(a|s) Q^{\pi_i}(s, a)$$

$$= (\epsilon/|A|) \sum_{a \in A} Q^{\pi_i}(s, a) + (1 - \epsilon) \max_a Q^{\pi_i}(s, a)$$

$$\pi_i(a|s) = \begin{cases} \epsilon/|A| \\ \text{Ret.} \end{cases}$$

$$\geq \text{“} + \sum_{a \in A} \left(\pi_i(a|s) - \frac{\epsilon}{|A|}\right) Q^{\pi_i}(s, a) = V^{\pi_i}(s)$$

Model-free control

Repeat:

- Sample episode $1, \ldots, k, \ldots$, using π: $\{S_1, A_1, R_2, \ldots, S_T\} \sim \pi$
- For each state S_t and action A_t in the episode,

$$q(S_t, A_t) \leftarrow q(S_t, A_t) + \alpha_t \left(G_t - q(S_t, A_t)\right)$$

- e.g.,

$$\alpha_t = \frac{1}{N(S_t, A_t)} \quad \text{of} \quad \alpha_t = 1/k$$

- Improve policy based on new action-value function

$$\epsilon \leftarrow 1/k$$
$$\pi \leftarrow \epsilon\text{-greedy}(q)$$

  - (Generalized the $\epsilon - greedy$ bandit algorithm)

GLIE

Greedy in the Limit with Infinite Exploration (GLIE)

- All state-action pairs are explored infinitely many times,

$$\forall s, a \quad \lim_{t \to \infty} N_t(s, a) = \infty$$

- The policy converges to a greedy policy,

$$\lim_{t \to \infty} \pi_t(a|s) = \mathcal{I}(a = \operatorname*{argmax}_{a'} q_t(s, a'))$$

- For example, $\epsilon - greedy$ with $\epsilon_k = \frac{1}{k}$

GLIE

Theorem

GLIE Monte-Carlo control converges to the optimal state-action value function $Q(s, a) \rightarrow Q^*(s, a)$

Monte Carlo Online Control / On Policy Improvement

1: Initialize $Q(s,a) = 0, N(s,a) = 0\ \forall (s,a)$, Set $\epsilon = 1$, $k = 1$
2: $\pi_k = \epsilon$-greedy(Q) // Create initial ϵ-greedy policy
3: **loop**
4: Sample k-th episode $(s_{k,1}, a_{k,1}, r_{k,1}, s_{k,2}, \ldots, s_{k,T})$ given π_k
4: $G_{k,t} = r_{k,t} + \gamma r_{k,t+1} + \gamma^2 r_{k,t+2} + \cdots \gamma^{T_i - 1} r_{k,T_i}$
5: **for** $t = 1, \ldots, T$ **do**
6: **if** First visit to (s,a) in episode k **then**
7: $N(s,a) = N(s,a) + 1$
8: $Q(s_t, a_t) = Q(s_t, a_t) + \frac{1}{N(s,a)}(G_{k,t} - Q(s_t, a_t))$
9: **end if**
10: **end for**
11: $k = k + 1$, $\epsilon = 1/k$
12: $\pi_k = \epsilon$-greedy(Q) // Policy improvement
13: **end loop**

Generated Data