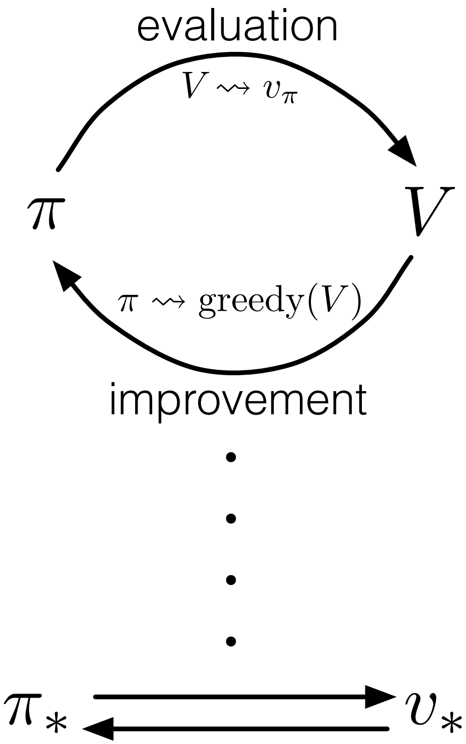
Chapter 5: Monte Carlo Methods

- Monte Carlo methods are learning methods

  Experience $\rightarrow$ values, policy

- Monte Carlo methods can be used in two ways:
  - *model-free:* No model necessary
  - *Simulated:* Needs only a simulation, not a *full* model
- Monte Carlo methods learn from *complete* sample returns
  - Only defined for episodic tasks (in this book)

High-level Ideas



Policy evaluation:

$$v_{k+1}(s) = \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a)\Big[r + \gamma v_k(s')\Big] \qquad \forall s \in \mathcal{S}$$

learn from ALL states: bootstrapping

Policy improvement:

$$\begin{aligned} \pi'(s) &= \arg\max_a q_\pi(s,a) \\ &= \arg\max_a \mathbb{E}[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s, A_t = a] \\ &= \arg\max_a \sum_{s',r} p(s',r|s,a)\Big[r + \gamma v_\pi(s')\Big], \end{aligned}$$

In DP, we assume $p(s', r \mid s, a)$ is known

In many practical problems, $p(s', r \mid s, a)$ is unknown

1) Policy evaluation (prediction): Estimate the values from trajectories
2) Policy improvement (control): Use estimated $q_\pi(s,a)$

Outline

- Monte Carlo Policy Evaluation (Prediction)
- Monte Carlo Policy Improvement (Control)
- Off-policy methods

Monte Carlo Policy Evaluation

- *Goal:* learn $v_\pi(s)$
- *Given:* some number of episodes under which contain *s*

$$S_0, A_0, R_1, S_1, A_1, R_2, \ldots, S_{T-1}, A_{T-1}, R_T$$

- *Idea:* Average returns observed after visits to *s*

$$G_t = R_{t+1} + \gamma R_{t+2} + \ldots + \gamma^{T-1} R_T$$

$$v_\pi(s) = \mathbb{E}_\pi[G_t \mid S_t = s]$$

Use sample mean obtained from episodes to estimate this

- *Every-Visit MC:* average returns for *every* time *s* is visited in an episode
- *First-visit MC:* average returns only for *first* time *s* is visited in an episode
- Both converge asymptotically

First-visit Monte Carlo Policy Evaluation

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 $Returns(S_t)$

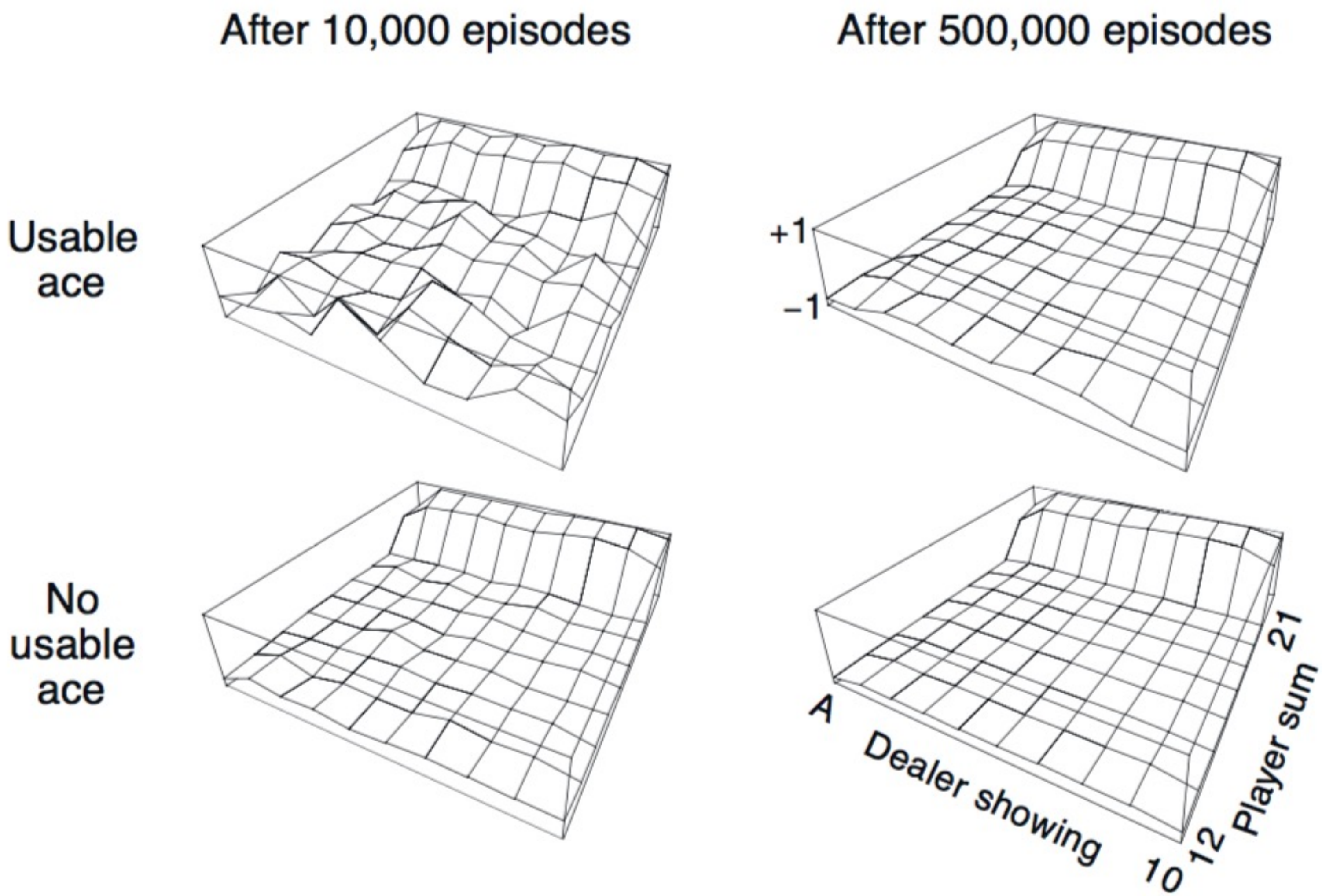
$V(S_t) \leftarrow \text{average}(Returns(S_t))$

Blackjack example

- *Object:* Have your card sum be greater than the dealer's without exceeding 21.
- *States* (200 of them):
  - current sum (12-21)
  - dealer's showing card (ace-10)
  - do I have a useable ace?

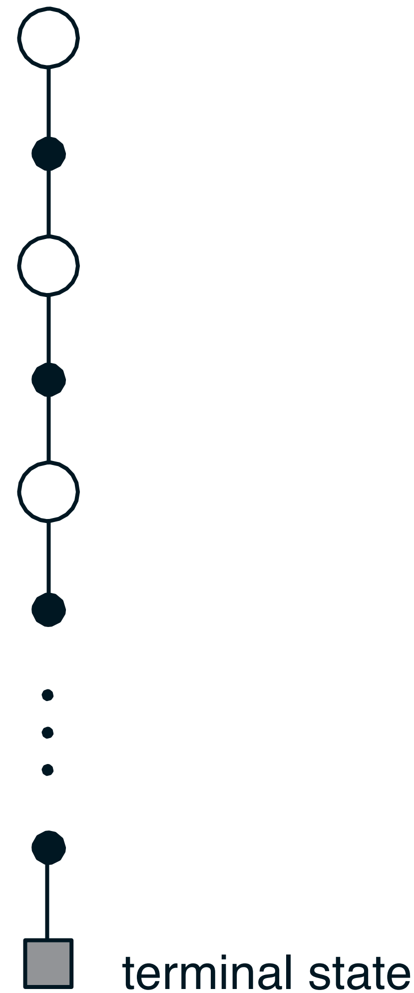
- *Reward:* $+1$ for winning, 0 for a draw, -1 for losing
- *Actions:* stick (stop receiving cards), hit (receive another card)
- *Policy:* Stick if my sum is 20 or 21, else hit
- No discounting ($\gamma = 1$)

Learned blackjack state-value functions



Backup diagram for Monte Carlo

- Entire rest of episode included
- Only one choice considered at each state (unlike DP)
  - thus, there will be an explore/exploit dilemma
- Does not bootstrap from successor states's values (unlike DP)
- Time required to estimate one state does not depend on the total number of states



Outline

- ❑ Monte Carlo Policy Evaluation (Prediction)
- ❑ Monte Carlo Policy Improvement (Control)
- ❑ Off-policy methods

Monte Carlo Estimation of Action Values (Q)

- State value not enough to pick an action when a model is not available

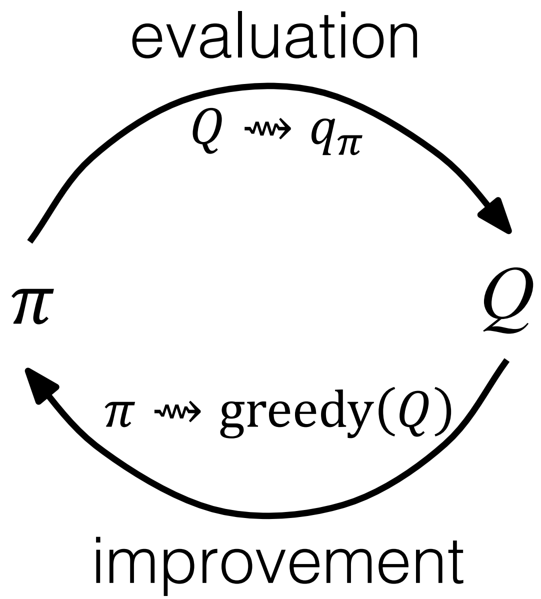
$$\begin{aligned}\pi'(s) &= \arg\max_a q_\pi(s,a) \\ &= \arg\max_a \mathbb{E}[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s, A_t = a] \\ &= \arg\max_a \sum_{s',r} p(s',r|s,a)\Big[r + \gamma v_\pi(s')\Big],\end{aligned}$$

- Monte Carlo is most useful when a model is not available

  We want to learn q_*

- $q_\pi(s,a)$ - average return starting from state *s* and action *a* following π
- Converges asymptotically *if* every state-action pair is visited
- *Exploring starts:* Every state-action pair has a non-zero probability of being the starting pair of an episode

Monte Carlo Control



- MC policy iteration: Policy evaluation using MC methods followed by policy improvement
- Policy improvement step: greedify with respect to value (or action-value) function

Convergence of MC Control

- Greedified policy meets the conditions for policy improvement:

$$\begin{aligned} q_{\pi_k}\big(s, \pi_{k+1}(s)\big) &= q_{\pi_k}\Big(s, \arg\max_a q_{\pi_k}(s,a)\Big) \\ &= \max_a q_{\pi_k}(s,a) \\ &\geq q_{\pi_k}\big(s, \pi_k(s)\big) \\ &= v_{\pi_k}(s) \end{aligned}$$

- And thus must be $\geq \pi_k$ by the policy improvement theorem
- This assumes exploring starts and infinite number of episodes for MC policy evaluation
- And :
  - update only to a given level of performance
  - alternate between evaluation and improvement per episode

Monte Carlo Exploring Starts

Monte Carlo ES (Exploring Starts), for estimating $\pi \approx \pi_*$

Initialize:

$\pi(s) \in \mathcal{A}(s)$ (arbitrarily), for all $s \in \mathcal{S}$

$Q(s, a) \in \mathbb{R}$ (arbitrarily), for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$

$Returns(s, a) \leftarrow$ empty list, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$

Loop forever (for each episode):

Choose $S_0 \in \mathcal{S}$, $A_0 \in \mathcal{A}(S_0)$ randomly such that all pairs have probability > 0

Generate an episode from S_0, A_0, 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 the pair S_t, A_t appears in $S_0, A_0, S_1, A_1 \ldots, S_{t-1}, A_{t-1}$:

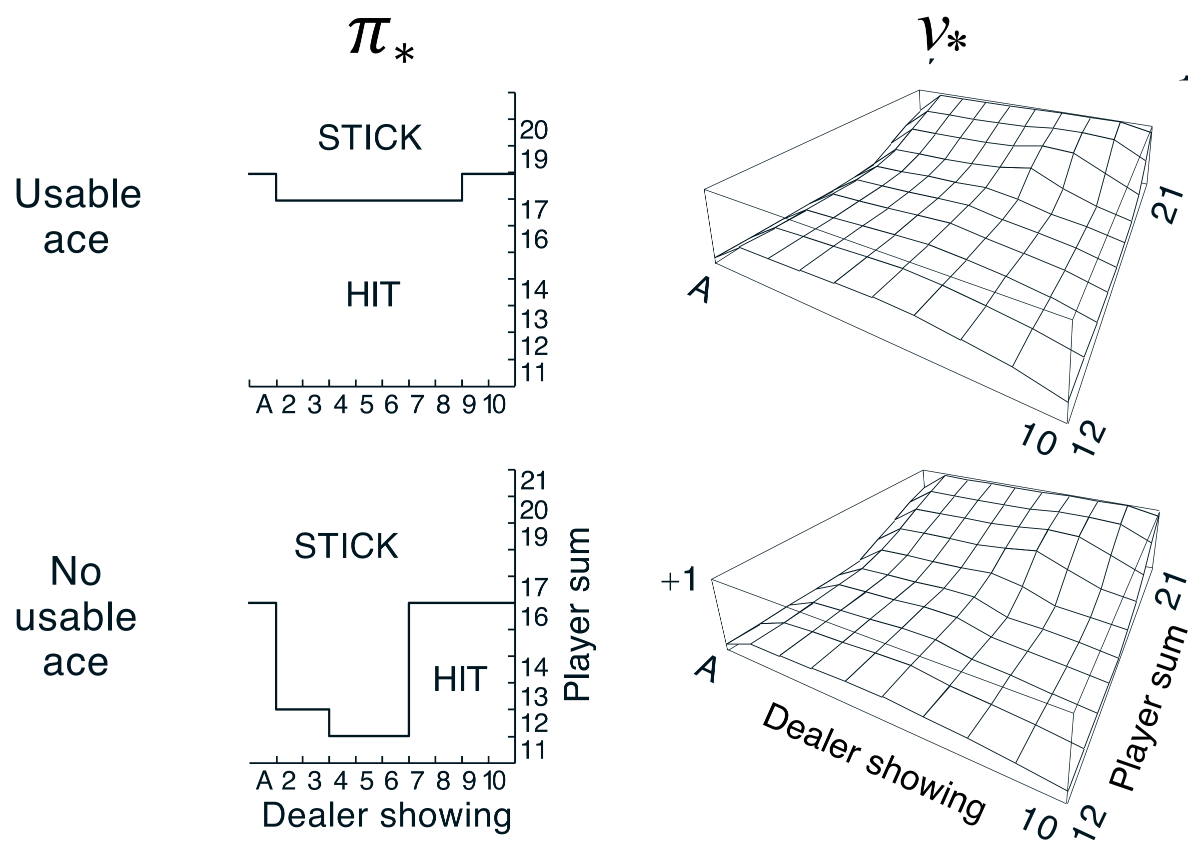
Append G to $Returns(S_t, A_t)$

$Q(S_t, A_t) \leftarrow \text{average}(Returns(S_t, A_t))$

$\pi(S_t) \leftarrow \arg\max_a Q(S_t, a)$

Blackjack example continued

- Exploring starts
- Initial policy as described before



On-policy Monte Carlo Control (for Exploration)

- *On-policy:* learn about policy currently executing
- How do we get rid of exploring starts?
  - The policy must be eternally *soft*:
    - $\pi(a|s) > 0$ for all s and a
  - e.g. ϵ - greedy policy:
    - probability of an action $= \dfrac{\epsilon}{|\mathcal{A}(s)|}$ or $1 - \epsilon + \dfrac{\epsilon}{|\mathcal{A}(s)|}$

      non-max max (greedy)

- Similar to GPI: move policy *towards* greedy policy (e.g., ϵ - greedy)
- Converges to best ϵ - soft policy

On-policy MC Control

On-policy first-visit MC control (for ε-soft policies), estimates $\pi \approx \pi_*$

Algorithm parameter: small $\varepsilon > 0$

Initialize:

  $\pi \leftarrow$ an arbitrary ε-soft policy

  $Q(s, a) \in \mathbb{R}$ (arbitrarily), for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$

  $Returns(s, a) \leftarrow$ empty list, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$

Repeat 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 the pair S_t, A_t appears in $S_0, A_0, S_1, A_1 \ldots, S_{t-1}, A_{t-1}$:

      Append G to $Returns(S_t, A_t)$

      $Q(S_t, A_t) \leftarrow \text{average}(Returns(S_t, A_t))$

      $A^* \leftarrow \arg\max_a Q(S_t, a)$ (with ties broken arbitrarily)

      For all $a \in \mathcal{A}(S_t)$:

$$\pi(a|S_t) \leftarrow \begin{cases} 1 - \varepsilon + \varepsilon/|\mathcal{A}(S_t)| & \text{if } a = A^* \\ \varepsilon/|\mathcal{A}(S_t)| & \text{if } a \neq A^* \end{cases}$$

What we’ve learned about Monte Carlo so far

- MC has several advantages over DP:
  - Can learn directly from interaction with environment
  - No need for full models
  - No need to learn from ALL states (no bootstrapping)
  - Less harmed by violating Markov property (later in book)
- MC methods provide an alternate policy evaluation process
- One issue to watch for: maintaining sufficient exploration
  - exploring starts, soft policies

Outline

- ❑ Monte Carlo Policy Evaluation (Prediction)
- ❑ Monte Carlo Policy Improvement (Control)
- ❑ Off-policy methods

Off-policy methods

- Learn the value of the *target policy* π from experience due to *behavior policy b*
- For example, π is the greedy policy (and ultimately the optimal policy) while b is exploratory (e.g., ϵ-soft)
- Why is this important?
  - Learn from human or other agents
  - Reuse trajectories produced by old policies
  - Learn about optimal policy while following exploratory policy
  - Learn about multiple policies while following one policy

High-level Idea: Importance Sampling

- In general, we only require *coverage*, i.e., that b generates behavior that covers, or includes, π

  $b(a|s) > 0$ for every s, a at which $\pi(a|s) > 0$

- Idea: *importance sampling*

  Suppose we want to compute the average according to distribution P, but we only have samples generated from Q

$$
\begin{aligned}
\mathbb{E}_{X\sim P}[g(X)] &= \sum P(X)g(X) \\
&= \sum Q(X)\frac{P(X)}{Q(X)}g(X) \\
&= \mathbb{E}_{X\sim Q}\left[g(X)\frac{P(X)}{Q(X)}\right]
\end{aligned}
$$

importance sampling ratio

Importance Sampling Ratio

- Probability of the rest of the trajectory, after S_t:

Policy π: $$Pr\{A_t, S_{t+1}, A_{t+1}, \cdots, S_T | S_t, A_{t:T-1} \sim \pi\}$$
$$= \pi(A_t|S_t)p(S_{t+1}|S_t, A_t)\pi(A_{t+1}|S_{t+1}) \cdots p(S_T|S_{T-1}, A_{T-1})$$
$$= \prod_{k=t}^{T-1} \pi(A_k|S_k)p(S_{k+1}|S_k, A_k)$$

Policy b: $$Pr\{A_t, S_{t+1}, A_{t+1}, \cdots, S_T | S_t, A_{t:T-1} \sim b\}$$
$$= b(A_t|S_t)p(S_{t+1}|S_t, A_t)b(A_{t+1}|S_{t+1}) \cdots p(S_T|S_{T-1}, A_{T-1})$$
$$= \prod_{k=t}^{T-1} b(A_k|S_k)p(S_{k+1}|S_k, A_k)$$

- In importance sampling, each return is weighted by the relative probability of the trajectory under the two policies

$$\rho_t^T = \frac{\prod_{k=t}^{T-1} \pi(A_k|S_k)p(S_{k+1}|S_k, A_k)}{\prod_{k=t}^{T-1} b(A_k|S_k)p(S_{k+1}|S_k, A_k)} = \prod_{k=t}^{T-1} \frac{\pi(A_k|S_k)}{b(A_k|S_k)}$$

Importance Sampling

- New notation: time steps increase across episode boundaries:

  - . . . s . . . . ▨ . . . . . . ▨ . . . s . . . . ▨ . . .
  - $t =$ 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27

$\mathcal{T}(s) = \{4,20\}$

set of start times

$T(4) = 9 \quad T(20) = 25$

next termination times

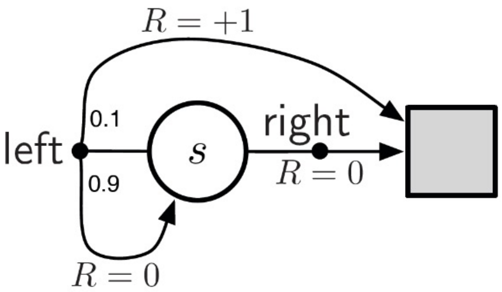
- *Ordinary importance sampling* forms estimate

$$V(s) \doteq \frac{\sum_{t\in\mathcal{T}(s)} \rho_t^{T(t)} G_t}{|\mathcal{T}(s)|}$$

- Whereas *weighted importance sampling* forms estimate (to reduce variance)

$$V(s) \doteq \frac{\sum_{t\in\mathcal{T}(s)} \rho_t^{T(t)} G_t}{\sum_{t\in\mathcal{T}(s)} \rho_t^{T(t)}}$$

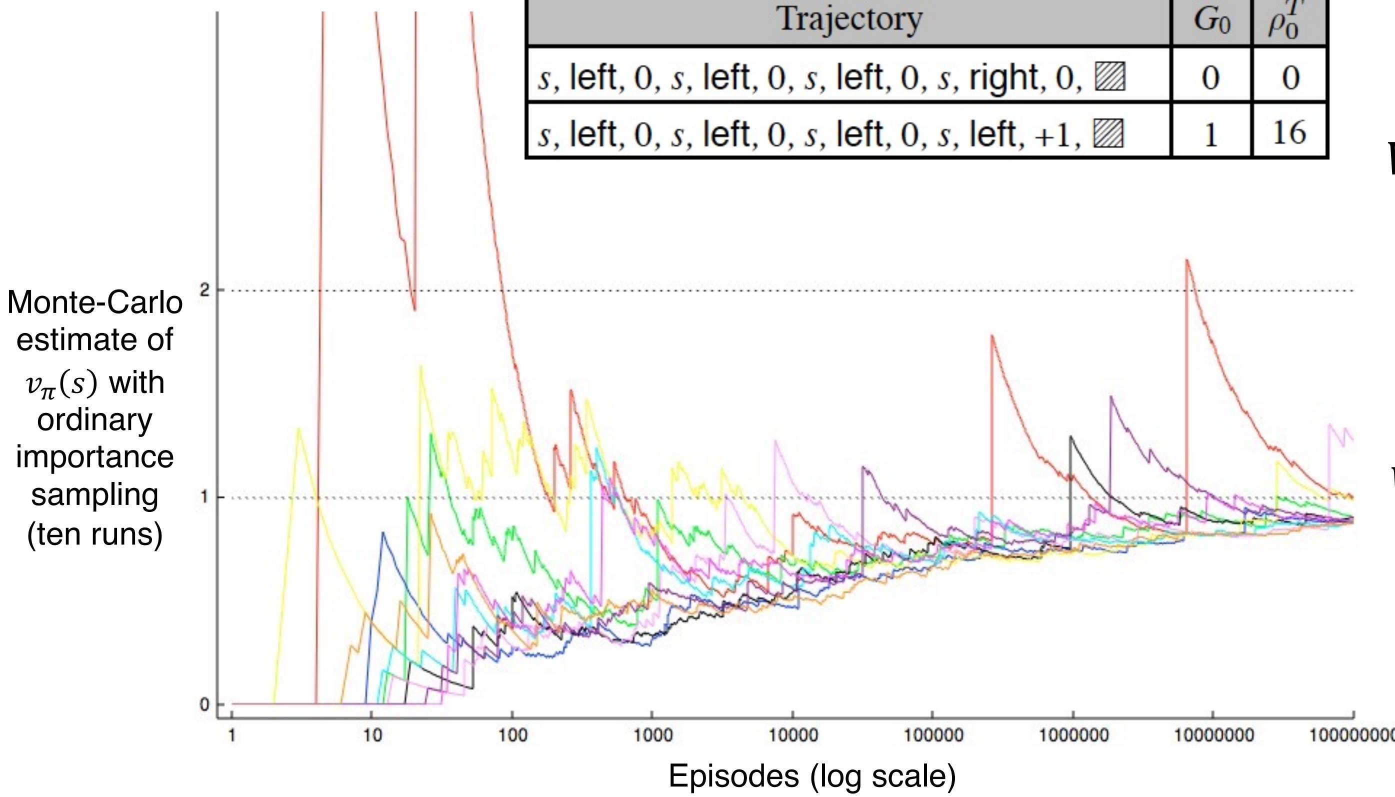
Example of infinite variance under *ordinary* importance sampling



$$\pi(\text{left}|s) = 1 \qquad \gamma = 1 \qquad \frac{\pi(\text{right}|s)}{b(\text{right}|s)} = 0 \qquad \frac{\pi(\text{left}|s)}{b(\text{left}|s)} = 2$$

$$b(\text{left}|s) = \frac{1}{2} \qquad v_\pi(s) = 1$$

Trajectory	G_0	ρ_0^T
s, left, 0, s, left, 0, s, left, 0, s, right, 0, ▨	0	0
s, left, 0, s, left, 0, s, left, 0, s, left, +1, ▨	1	16



OIS:

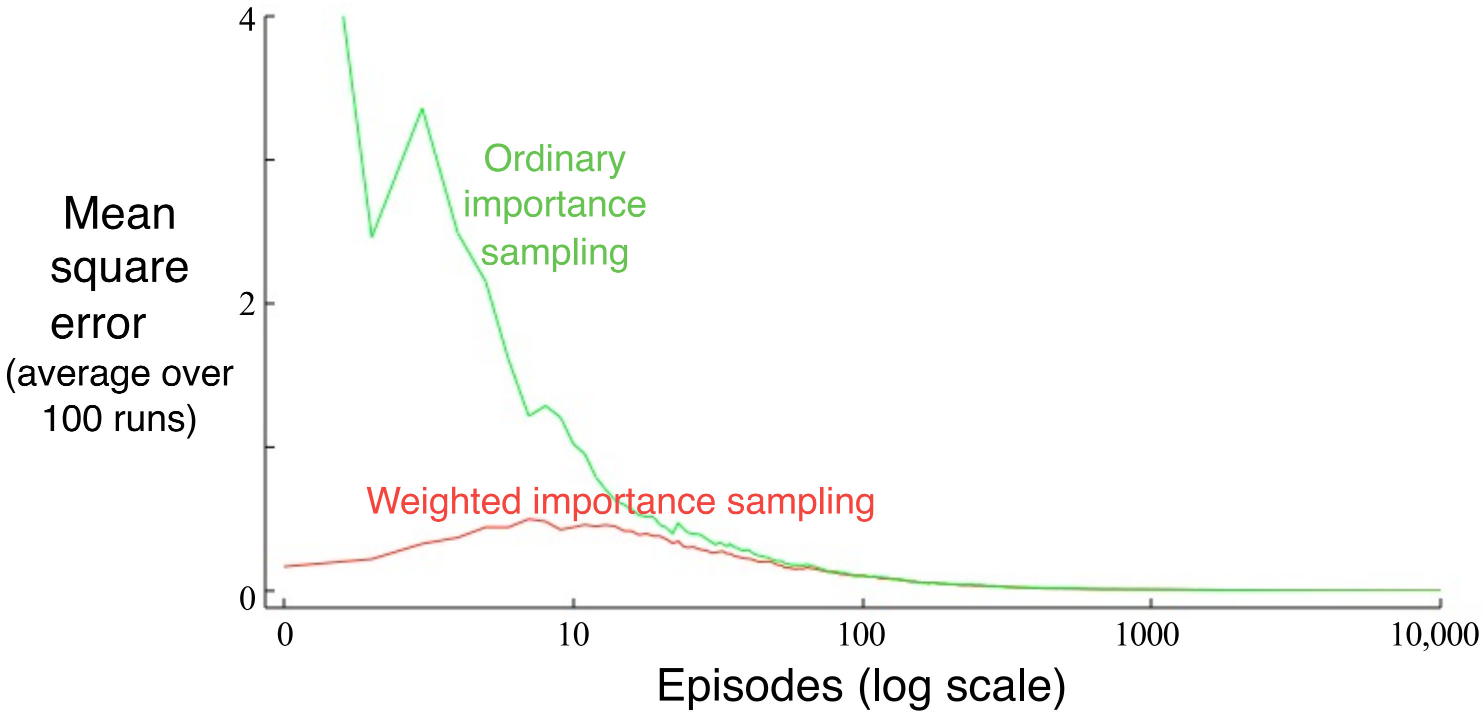
$$V(s) \doteq \frac{\sum_{t\in\mathcal{T}(s)} \rho_t^{T(t)} G_t}{|\mathcal{T}(s)|}$$

WIS:

$$V(s) \doteq \frac{\sum_{t\in\mathcal{T}(s)} \rho_t^{T(t)} G_t}{\sum_{t\in\mathcal{T}(s)} \rho_t^{T(t)}}$$

Example: Off-policy Estimation of the value of a *single* Blackjack State

- State is player-sum 13, dealer-showing 2, useable ace
- Target policy is stick only on 20 or 21
- Behavior policy is equiprobable
- True value ≈ −0.27726



Incremental off-policy every-visit MC policy evaluation (returns $Q \approx q_\pi$)

Input: an arbitrary target policy π

Initialize, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$:
$Q(s, a) \leftarrow$ arbitrary
$C(s, a) \leftarrow 0$

Repeat forever:
$b \leftarrow$ any policy with coverage of π
Generate an episode using μ:
$S_0, A_0, R_1, \cdots S_{T-1}, A_{T-1}, R_T, S_T$
$G \leftarrow 0$
$W \leftarrow 1$
For $t = T-1, T-2, \cdots$ downto 0:
$G \leftarrow \gamma G + R_{t+1}$
$C(S_t, A_t) \leftarrow C(S_t, A_t) + W$
$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \frac{W}{C(S_t, A_t)}[G - Q(S_t, A_t)]$
$W \leftarrow W \dfrac{\pi(A_t|S_t)}{b(A_t|S_t)}$
If $W = 0$ then ExitForLoop

Off-policy every-visit MC control (returns $\pi \approx \pi_*$)

Initialize, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$:

$Q(s, a) \leftarrow$ arbitrary

$C(s, a) \leftarrow 0$

$\pi(s) \leftarrow \arg\max_a Q(S_t, a)$ (with ties broken consistently)

Repeat forever:

$b \leftarrow$ any soft policy

Generate an episode using μ:

$S_0, A_0, R_1, \cdots S_{T-1}, A_{T-1}, R_T, S_T$

$G \leftarrow 0$

$W \leftarrow 1$

For $t = T-1, T-2, \cdots$ downto 0:

$G \leftarrow \gamma G + R_{t+1}$

$C(S_t, A_t) \leftarrow C(S_t, A_t) + W$

$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \frac{W}{C(S_t, A_t)}[G - Q(S_t, A_t)]$

$\pi(S_t) \leftarrow \arg\max_a Q(S_t, a)$ (with ties broken consistently)

If $A_t \neq \pi(S_t)$ then ExitForLoop

$W \leftarrow W \dfrac{1}{b(A_t|S_t)}$

Target policy is greedy and deterministic

Behavior policy is soft, typically ϵ-greedy

Summary

- MC has several advantages over DP:
  - Can learn directly from interaction with environment
  - No need for full models
  - Less harmed by violating Markov property (later in book)
- MC methods provide an alternate policy evaluation process
- One issue to watch for: maintaining sufficient exploration
  - exploring starts, soft policies
- Introduced distinction between *on-policy* and *off-policy* methods
- Introduced *importance sampling* for off-policy learning
- Introduced distinction between *ordinary* and *weighted* IS