Chapter 5: Monte Carlo Methods

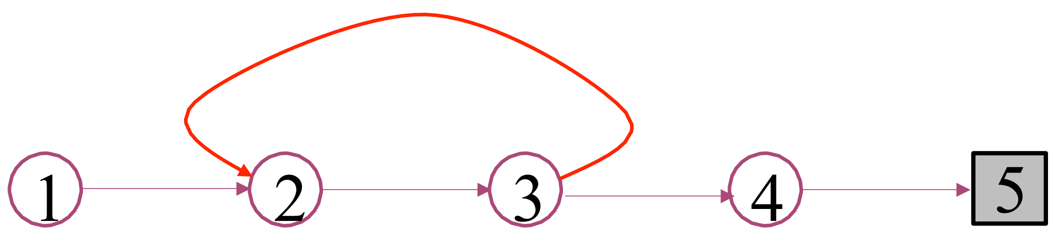
- Monte Carlo methods are learning methods

  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)
- Like an associative version of a bandit method

Monte Carlo Policy Evaluation

- *Goal:* learn $v_\pi(s)$
- *Given:* some number of episodes under π which contain *s*
- *Idea:* Average returns observed after visits to *s*



- *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

Initialize:

$\pi \leftarrow$ policy to be evaluated

$V \leftarrow$ an arbitrary state-value function

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

Repeat forever:

Generate an episode using π

For each state s appearing in the episode:

$G \leftarrow$ return following the first occurrence of s

Append G to $Returns(s)$

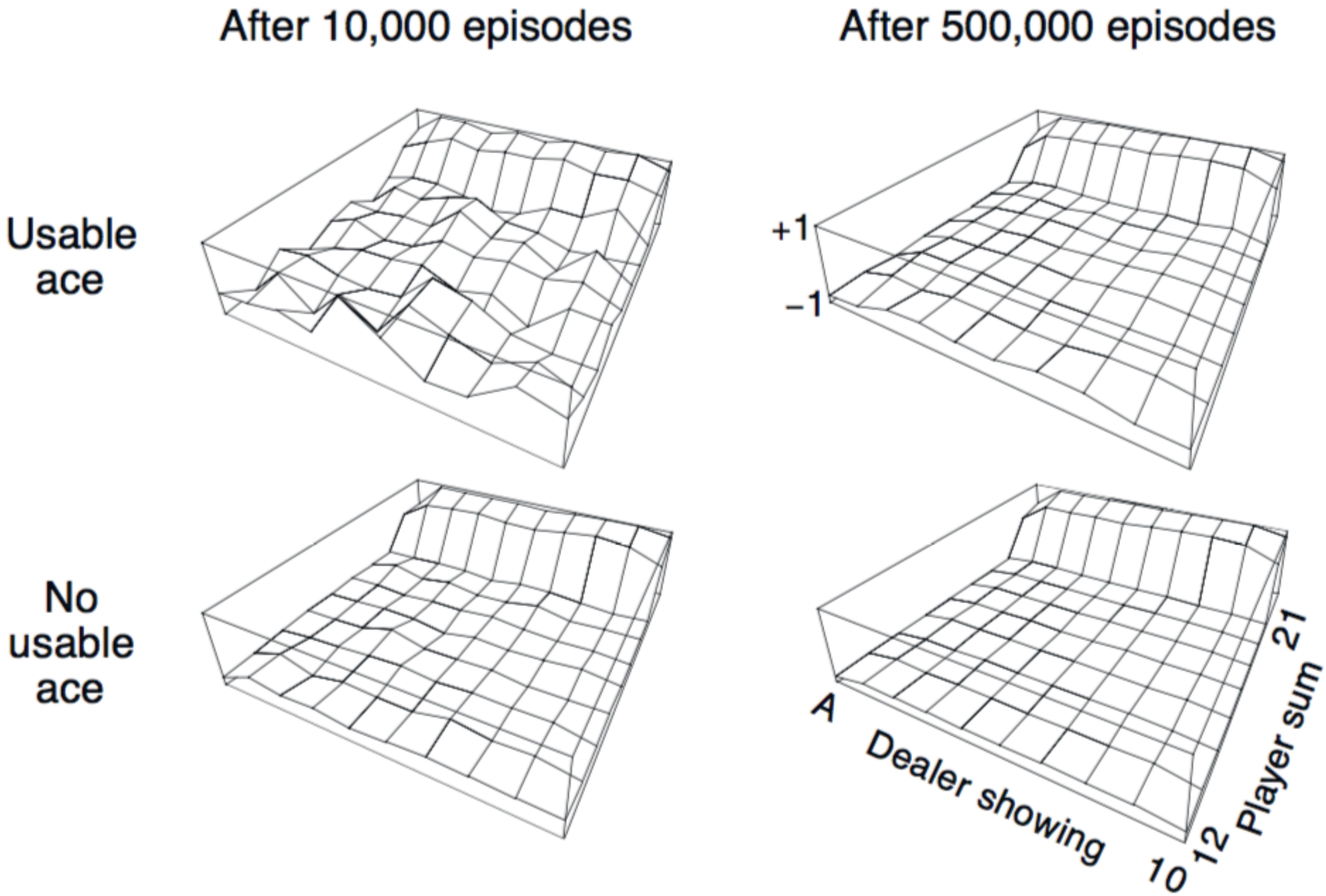
$V(s) \leftarrow$ average $\left(Returns(s)\right)$

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?

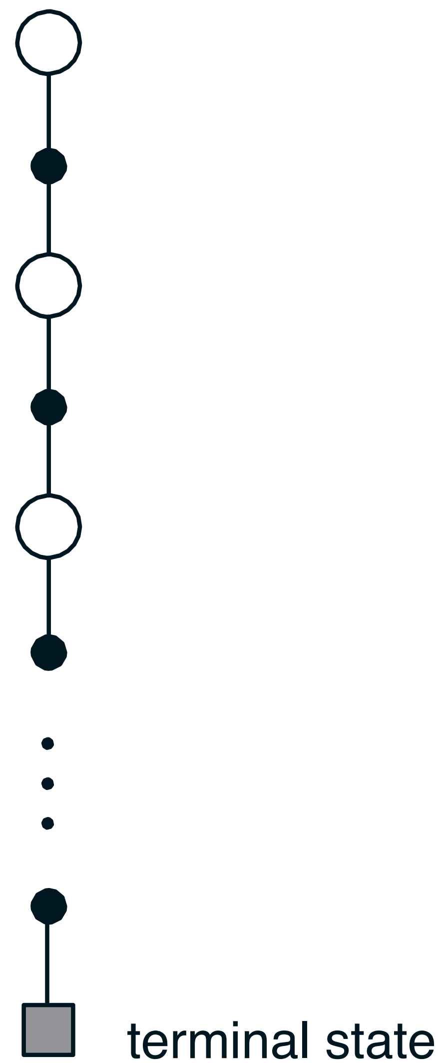
- *Reward:* +1 for winning, 0 for a draw, −1 for losing
- *Actions:* stick (stop receiving cards), hit (receive another card)
- *Policy:* Stick if my sum is 20 or 21, else hit
- No discounting ($\gamma = 1$)

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

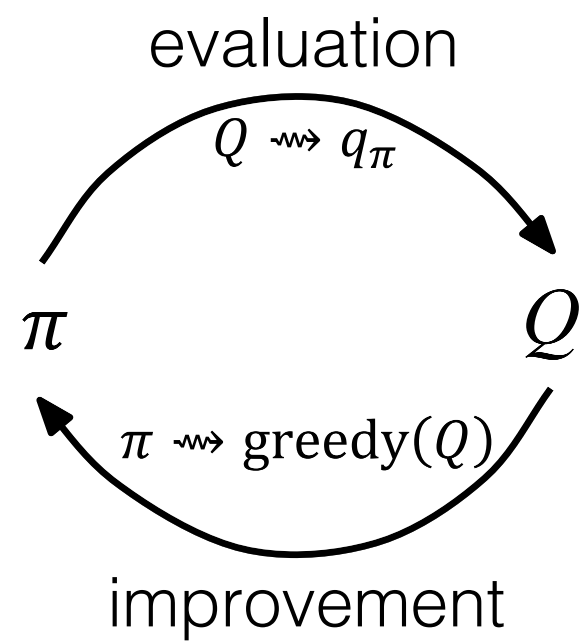


Monte Carlo Estimation of Action Values (Q)

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

- 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

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:

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

- 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

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

$Q(s, a) \leftarrow$ arbitrary

$\pi(s) \leftarrow$ arbitrary

$Returns(s, a) \leftarrow$ empty list

Fixed point is optimal policy π^*

Being Proven (almost)

Repeat forever:

Choose $S_0 \in \mathcal{S}$ and $A_0 \in \mathcal{A}(S_0)$ s.t. all pairs have probability > 0

Generate an episode starting from S_0, A_0, following π

For each pair s, a appearing in the episode:

$G \leftarrow$ return following the first occurrence of s, a

Append G to $Returns(s, a)$

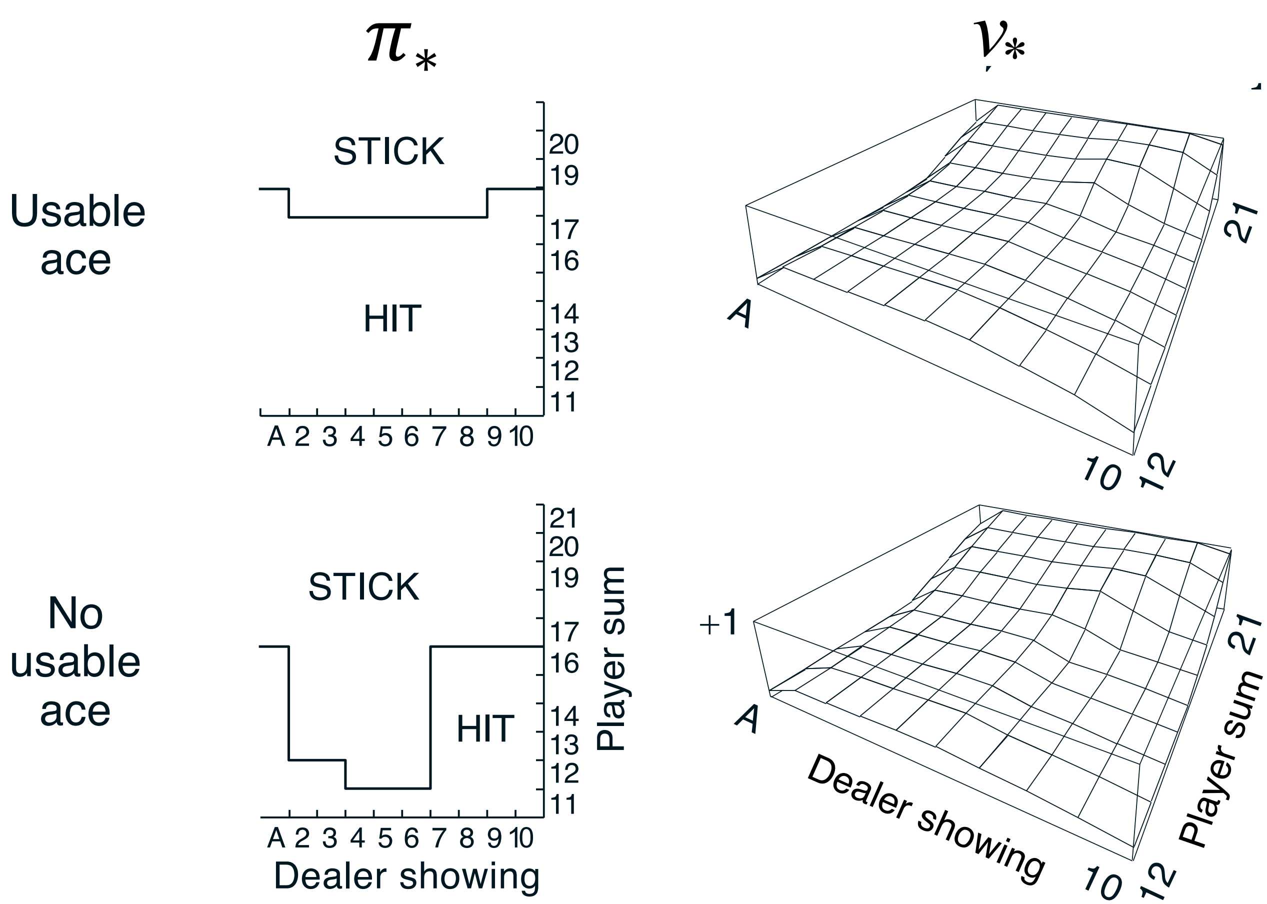
$Q(s, a) \leftarrow \text{average}\big(Returns(s, a)\big)$

For each s in the episode:

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

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

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

$Q(s, a) \leftarrow$ arbitrary

$Returns(s, a) \leftarrow$ empty list

$\pi(a|s) \leftarrow$ an arbitrary ϵ-soft policy

Repeat forever:

(a) Generate an episode using π

(b) For each pair s, a appearing in the episode:

$G \leftarrow$ return following the first occurrence of s, a

Append G to $Returns(s, a)$

$Q(s, a) \leftarrow$ average($Returns(s, a)$)

(c) For each s in the episode:

$A^* \leftarrow \arg\max_a Q(s, a)$

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

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

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

Off-policy methods

- Learn the value of the *target policy* π from experience due to *behavior policy* μ
- For example, π is the greedy policy (and ultimately the optimal policy) while μ is exploratory (e.g., ϵ-soft)
- In general, we only require *coverage*, i.e., that μ generates behavior that covers, or includes, π

$$\mu(a|s) > 0 \text{ for every } s, a \text{ at which } \pi(a|s) > 0$$

- Idea: *importance sampling*
  - Weight each return by the *ratio of the probabilities* of the trajectory under the two policies

Importance Sampling Ratio

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

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

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

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

- This is called the *importance sampling ratio*
- All importance sampling ratios have expected value 1→unbiased

$$\mathbb{E}_{A_k \sim \mu}\left[\frac{\pi(A_k|S_k)}{\mu(A_k|S_k)}\right] = \sum_a \mu(a|S_k) \frac{\pi(a|S_k)}{\mu(a|S_k)} = \sum_a \pi(a|S_k) = 1$$

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

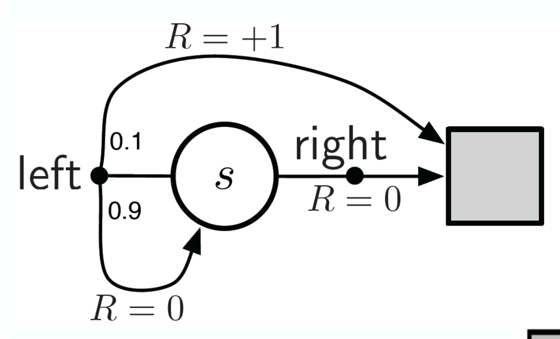
- *Ordinary importance sampling* forms estimate

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

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

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

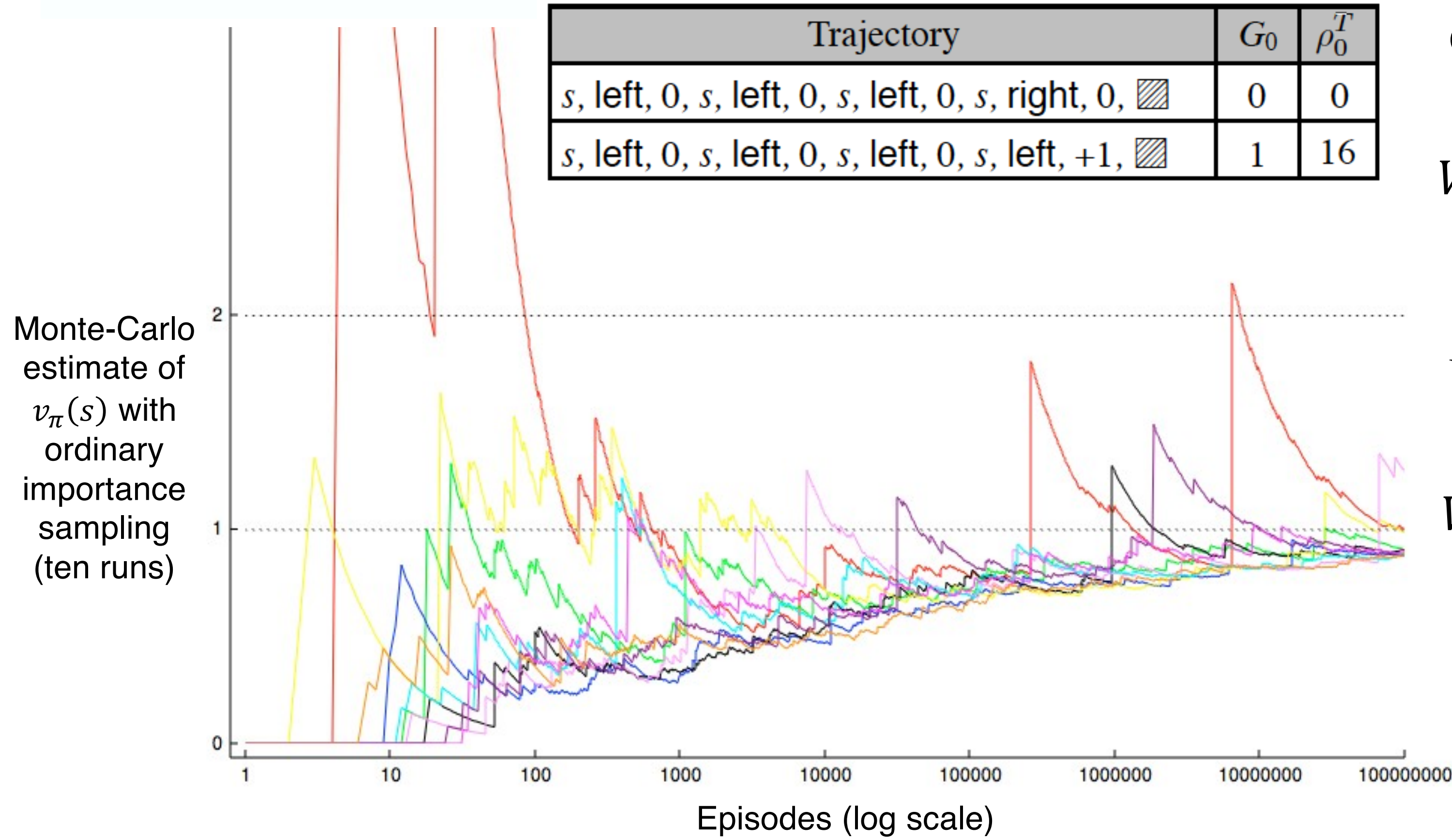
Example of infinite variance under *ordinary* importance sampling



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

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

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:

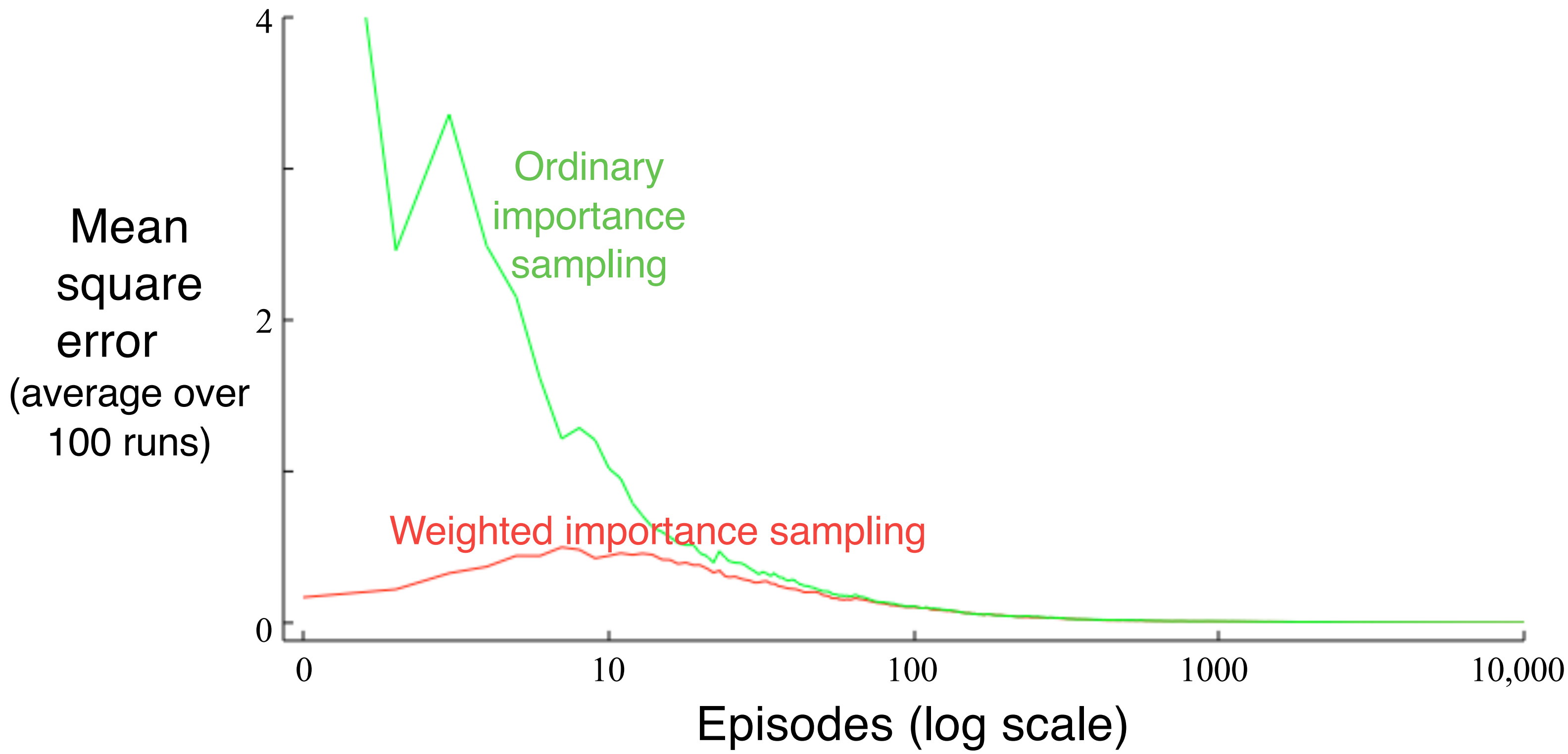
$$V(s) \doteq \frac{\sum_{t \in \mathcal{T}(s)} \rho_t^{T(t)} G_t}{|\mathcal{T}(s)|}$$

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:
$\mu \leftarrow$ 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:
$G \leftarrow \gamma G + R_{t+1}$
$C(S_t, A_t) \leftarrow C(S_t, A_t) + W$
$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \frac{W}{C(S_t, A_t)}[G - Q(S_t, A_t)]$
$W \leftarrow W \frac{\pi(A_t|S_t)}{\mu(A_t|S_t)}$
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:

$\mu \leftarrow$ 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

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

Target policy is greedy and deterministic

Behavior policy is soft, typically ϵ-greedy

Discounting-aware Importance Sampling (motivation)

- So far we have weighted returns without taking into account that they are a discounted sum
- This can't be the best one can do!
- For example, suppose $\gamma = 0$
  - Then G_0 will be weighted by

$$\rho_t^T = \frac{\pi(A_0|S_0)}{\mu(A_0|S_0)}\frac{\pi(A_1|S_1)}{\mu(A_1|S_1)}\cdots\frac{\pi(A_{T-1}|S_{T-1})}{\mu(A_{T-1}|S_{T-1})}$$

  - But it really need only be weighted by

$$\rho_t^1 = \frac{\pi(A_0|S_0)}{\mu(A_0|S_0)}$$

  - Which would have <u>much smaller variance</u>

Discounting-aware Importance Sampling

- Define the flat partial return:

$$\bar{G}_t^h \triangleq R_{t+1} + R_{t+2} + \cdots + R_h, \qquad 0 \le t < h \le T,$$

- Then

$$\begin{aligned}
G_t &\triangleq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots + \gamma^{T-t-1} R_T \\
&= (1-\gamma) R_{t+1} \\
&\quad + (1-\gamma)\gamma (R_{t+1} + R_{t+2}) \\
&\quad + (1-\gamma)\gamma^2 (R_{t+1} + R_{t+2} + R_{t+3}) \\
&\quad \vdots \\
&\quad + (1-\gamma)\gamma^{T-t-2} (R_{t+1} + R_{t+2} + \cdots + R_{T-1}) \\
&\quad + \gamma^{T-t-1} (R_{t+1} + R_{t+2} + \cdots + R_T) \\
&= (1-\gamma) \sum_{h=t+1}^{T-1} \gamma^{h-t-1} \bar{G}_t^h + \gamma^{T-t-1} \bar{G}_t^T
\end{aligned}$$

Discounting-aware Importance Sampling

- Define the flat partial return:

$$\bar{G}_t^h \triangleq R_{t+1} + R_{t+2} + \cdots + R_h, \qquad 0 \leq t < h \leq T,$$

- Then

$$G_t = (1-\gamma) \sum_{h=t+1}^{T-1} \gamma^{h-t-1} \bar{G}_t^h + \gamma \bar{G}_t^T$$

- Ordinary discounting-aware IS:

$$V(s) \triangleq \frac{\sum_{t \in \mathcal{T}(s)} \left((1-\gamma) \sum_{h=t+1}^{T(t)-1} \gamma^{h-t-1} \rho_t^h \bar{G}_t^h + \gamma^{T(t)-t-1} \rho_t^{T(t)} \bar{G}_t^{T(t)} \right)}{|\mathcal{T}(s)|}$$

- Weighted discounting-aware IS:

$$V(s) \triangleq \frac{\sum_{t \in \mathcal{T}(s)} \left((1-\gamma) \sum_{h=t+1}^{T(t)-1} \gamma^{h-t-1} \rho_t^h \bar{G}_t^h + \gamma^{T(t)-t-1} \rho_t^{T(t)} \bar{G}_t^{T(t)} \right)}{\sum_{t \in \mathcal{T}(s)} \left((1-\gamma) \sum_{h=t+1}^{T(t)-1} \gamma^{h-t-1} \rho_t^h + \gamma^{T(t)-t-1} \rho_t^{T(t)} \right)}$$

Per-reward Importance Sampling

- Another way of reducing variance, even if $\gamma = 1$
- Uses the fact that the return is a *sum of rewards*

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

- where

$$\rho_t^T R_{t+k} = \frac{\pi(A_t|S_t)}{\mu(A_t|S_t)} \frac{\pi(A_{t+1}|S_{t+1})}{\mu(A_{t+1}|S_{t+1})} \cdots \cancel{\frac{\pi(A_{t+k}|S_{t+k})}{\mu(A_{t+k}|S_{t+k})} \cdots \frac{\pi(A_{T-1}|S_{T-1})}{\mu(A_{T-1}|S_{T-1})}} R_{t+k}$$

$$\therefore \mathbb{E}[\rho_t^T R_{t+k}] = \mathbb{E}\left[\rho_t^{t+k} R_{t+k}\right]$$

$$\therefore \mathbb{E}[\rho_t^T G_t] = \mathbb{E}\,\underbrace{\left[\rho_t^{t+1} R_{t+1} + +\gamma \rho_t^{t+2} R_{t+2} + \gamma^2 \rho_t^{t+3} R_{t+3} + \cdots + \gamma^{T-t-1} \rho_t^T R_T\right]}_{\tilde{G}_t}$$

- Per-reward ordinary IS:

$$V(s) \triangleq \frac{\sum_{t \in \mathcal{T}(s)} \tilde{G}_t}{|\mathcal{T}(s)|}$$

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
- Introduced two *return-specific* ideas for reducing IS variance
  - *discounting-aware* and *per-reward* IS