Monte Carlo Control

CMPUT 366: Intelligent Systems

S&B §5.3-5.5, 5.7

Lecture Outline

1. Recap
2. Estimating Action Values
3. Monte Carlo Control
4. Importance Sampling
5. Off-Policy Monte Carlo Control

Assignment #3

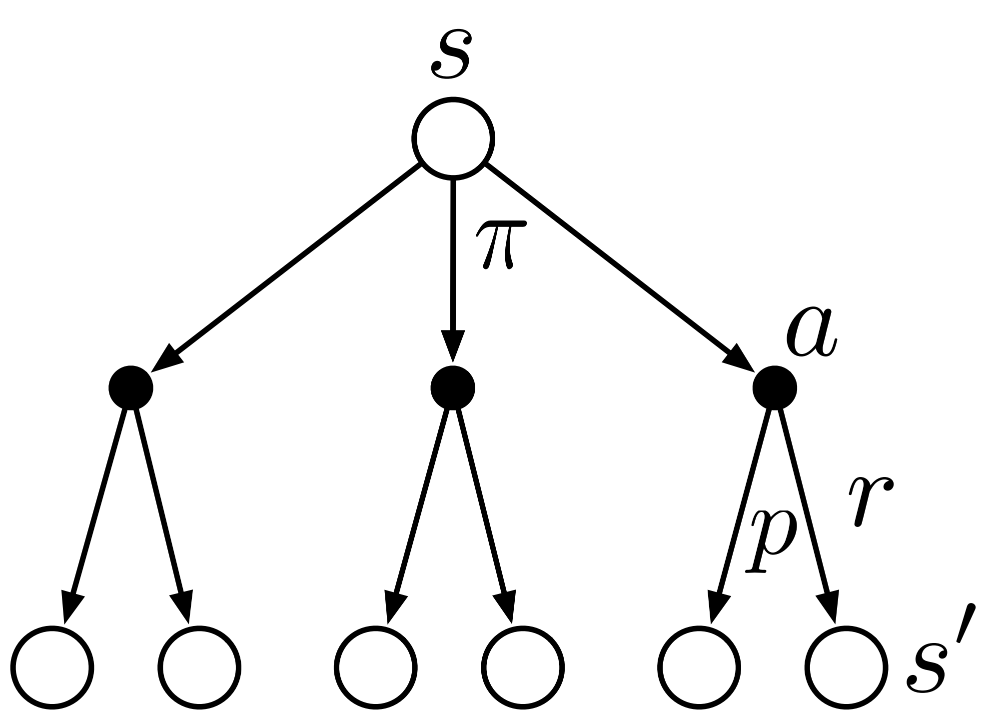
- Assignment #3 is **due Friday** (Mar 25) at **11:59pm**
  - This is a firm deadline
  - TAs have office hours every day this week

Recap: Policy Evaluation

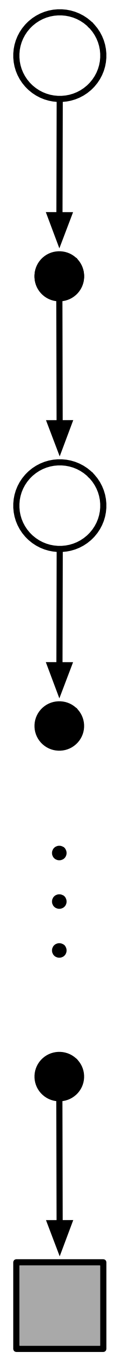
Question: How can we **compute** v_π?

1. We know that v_π is the unique solution to the Bellman equations, so we could just **solve them** (treating $v_\pi(s_1), \ldots, v_\pi(s_{|\mathcal{S}|})$ as variables)
   - but that is tedious and annoying and slow
     ~~(it's a system of $|\mathcal{S}|$ nonlinear equations)~~
     (it's a system of $|\mathcal{S}|$ **linear** equations in $|\mathcal{S}|$ unknowns)
     $$v_\pi(s) = \sum_a \pi(a \mid s) \sum_{s',r} p(s', r \mid s, a)[r + \gamma v_\pi(s')]$$
   - Also requires a complete model of the dynamics
2. **Iterative policy evaluation**
   - Takes advantage of the recursive formulation

Recap: Monte Carlo vs. Dynamic Programming



- **Iterative policy evaluation** uses the estimates of the **next state's** value to update the value of this state
    - Only needs to compute a **single transition** to update a state's estimate
- **Monte Carlo** estimate of each state's value is **independent** from estimates of **other states'** values
    - Needs the **entire episode** to compute an update
    - Can focus on evaluating a **subset of states** if desired



First-visit Monte Carlo Prediction

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 $Returns(S_t)$

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

Control vs. Prediction

- **Prediction:** estimate the value of states and/or actions given some fixed policy π
- **Control:** estimate an optimal policy

Estimating Action Values

- When we know the **dynamics** $p(s', r \mid s, a)$, an estimate of **state values** is sufficient to determine a good **policy**:
    - Choose the action that gives the best combination of **reward** and **next-state value**:

$$\hat{a}^* = \arg\min_{a \in \mathscr{A}} \sum_{s',r} p(s', r \mid s, a)[r + \gamma \hat{v}(s')]$$

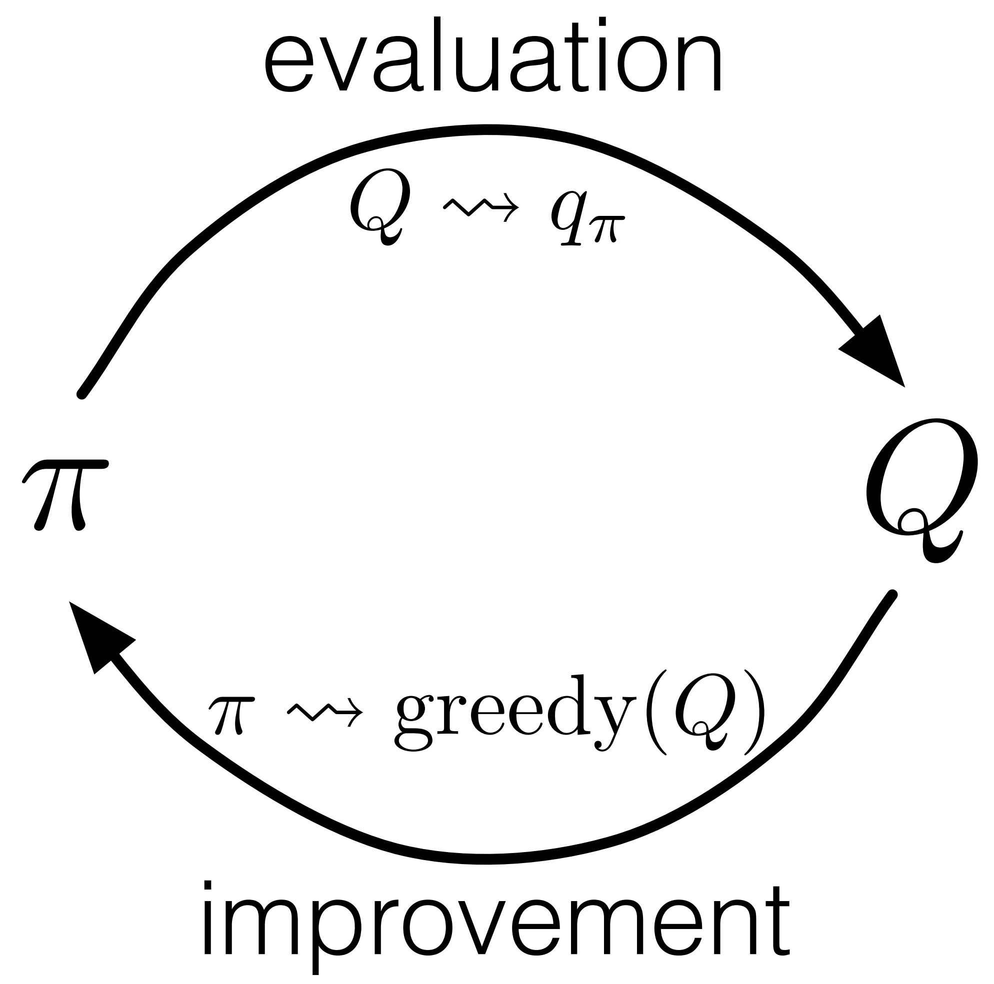
- If we don't know the dynamics, state values are **not enough**
    - To estimate a good policy, we need an **explicit** estimate of **action values**

Exploring Starts

- We can just run first-visit Monte Carlo and approximate the returns to each **state-action pair**
- **Question:** What do we do about state-action pairs that are **never visited**?
  - If the current policy π never selects an action a from a state s, then Monte Carlo can't estimate its value
- **Exploring starts assumption:**
  - Every episode **starts** at a random state-action pair S_0, A_0
  - **Every pair** has a positive probability of being selected for a start

Monte Carlo Control

Monte Carlo control can be used for **policy iteration**:



$$\pi_0 \xrightarrow{\mathrm{E}} q_{\pi_0} \xrightarrow{\mathrm{I}} \pi_1 \xrightarrow{\mathrm{E}} q_{\pi_1} \xrightarrow{\mathrm{I}} \pi_2 \xrightarrow{\mathrm{E}} \cdots \xrightarrow{\mathrm{I}} \pi_* \xrightarrow{\mathrm{E}} q_*$$

Monte Carlo Control with 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 $Returns(S_t, A_t)$
    - $Q(S_t, A_t) \leftarrow \text{average}(Returns(S_t, A_t))$
    - $\pi(S_t) \leftarrow \arg\max_a Q(S_t, a)$

Question: What unlikely assumptions does this rely upon?

ε-Soft Policies

- The **exploring starts** assumption requires that we see **every** state-action pair with positive probability
  - Even if π **never** chooses a from state s
- Another approach: Simply **force** π to (sometimes) choose a!
- An **ϵ-soft policy** is one for which $\pi(a \mid s) \geq \dfrac{\epsilon}{|\mathcal{A}(s)|} \quad \forall s, a$
- **Example: ϵ-greedy policy**

$$\pi(a \mid s) = \begin{cases} \frac{\epsilon}{|\mathcal{A}|} & \text{if } a \notin \arg\max_a Q(s,a), \\ (1-\epsilon) + \frac{\epsilon}{|\mathcal{A}|} & \text{otherwise.} \end{cases}$$

Monte Carlo Control w/out Exploring Starts

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

Algorithm parameter: small $\varepsilon > 0$

Initialize:

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

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

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

Repeat forever (for each episode):

$\quad$ Generate an episode following π: $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$

$\quad G \leftarrow 0$

$\quad$ Loop for each step of episode, $t = T-1, T-2, \ldots, 0$:

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

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

$\quad\quad\quad$ Append G to $\mathit{Returns}(S_t, A_t)$

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

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

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

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

Monte Carlo Control w/out Exploring Starts

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

Algorithm parameter: small $\varepsilon > 0$

Initialize:

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

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

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

Repeat forever (for each episode):

$\quad$ Generate an episode following π: $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$

$\quad G \leftarrow 0$

$\quad$ Loop for each step of episode, $t = T-1, T-2, \ldots, 0$:

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

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

$\quad\quad\quad$ Append G to $Returns(S_t, A_t)$

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

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

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

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

Question:

Will this procedure converge to the **optimal** policy π^*?

Why or why not?

Importance Sampling

- **Monte Carlo sampling:** use samples from the **target** distribution to estimate expectations
- **Importance sampling:** Use samples from **proposal** distribution to estimate expectations of **target** distribution by **reweighting** samples

$$\mathbb{E}[X] = \sum_x f(x)x = \sum_x \frac{g(x)}{g(x)} f(x)x = \sum_x g(x)\frac{f(x)}{g(x)}x \approx \frac{1}{n}\sum_{x_i \sim g} \boxed{\frac{f(x_i)}{g(x_i)}} x_i$$

Importance sampling ratio

Off-Policy Prediction via Importance Sampling

Definition:
Off-policy learning means using data generated by a **behaviour policy** to learn about a distinct **target policy**.

Proposal distribution

Target distribution

Off-Policy Monte Carlo Prediction

- Generate episodes using **behaviour policy** b
- Take **weighted** average of returns to state s over all the episodes containing a visit to s to estimate $v_\pi(s)$
  - Weighed by **importance sampling ratio** of trajectory starting from $S_t = s$ until the end of the episode:

$$\rho_{t:T-1} \doteq \frac{\Pr[A_t, S_{t+1}, \ldots, S_T \mid S_t, A_{t:T-1} \sim \pi]}{\Pr[A_t, S_{t+1}, \ldots, S_T \mid S_t, A_{t:T-1} \sim b]}$$

Importance Sampling Ratios for Trajectories

- **Probability of a trajectory** $A_t, S_{t+1}, A_{t+1}, \dots, S_T$ from S_t:

$$\Pr[A_t, S_{t+1}, \dots, S_T \mid S_t, A_{t:T-1} \sim \pi] =$$
$$\pi(A_t \mid S_t) p(S_{t+1} \mid S_t, A_t) \pi(A_{t+1} \mid S_{t+1}) \dots p(S_T \mid S_{T-1}, A_{T-1})$$

- **Importance sampling ratio** for a trajectory $A_t, S_{t+1}, A_{t+1}, \dots, S_T$ from S_t:

$$\rho_{t:T-1} \doteq \frac{\prod_{k=t}^{T-1} \pi(A_k \mid S_k) \boxed{p(S_{k+1} \mid S_k, A_k)}}{\prod_{k=t}^{T-1} b(A_k \mid S_k) \boxed{p(S_{k+1} \mid S_k, A_k)}} = \frac{\prod_{k=t}^{T-1} \pi(A_k \mid S_k)}{\prod_{k=t}^{T-1} b(A_k \mid S_k)}$$

$= 1$

Ordinary vs.Weighted Importance Sampling

- **Ordinary importance sampling:**

$$V(s) \doteq \frac{1}{n} \sum_{i=1}^{n} \rho_{t(s,i):T(i)-1} G_{i,t}$$

- **Weighted importance sampling:**

$$V(s) \doteq \frac{\sum_{i=1}^{n} \rho_{t(s,i):T(i)-1} G_{i,t}}{\sum_{i=1}^{n} \rho_{t(s,i):T(i)-1}}$$

Example: Ordinary vs. Weighted Importance Sampling for Blackjack

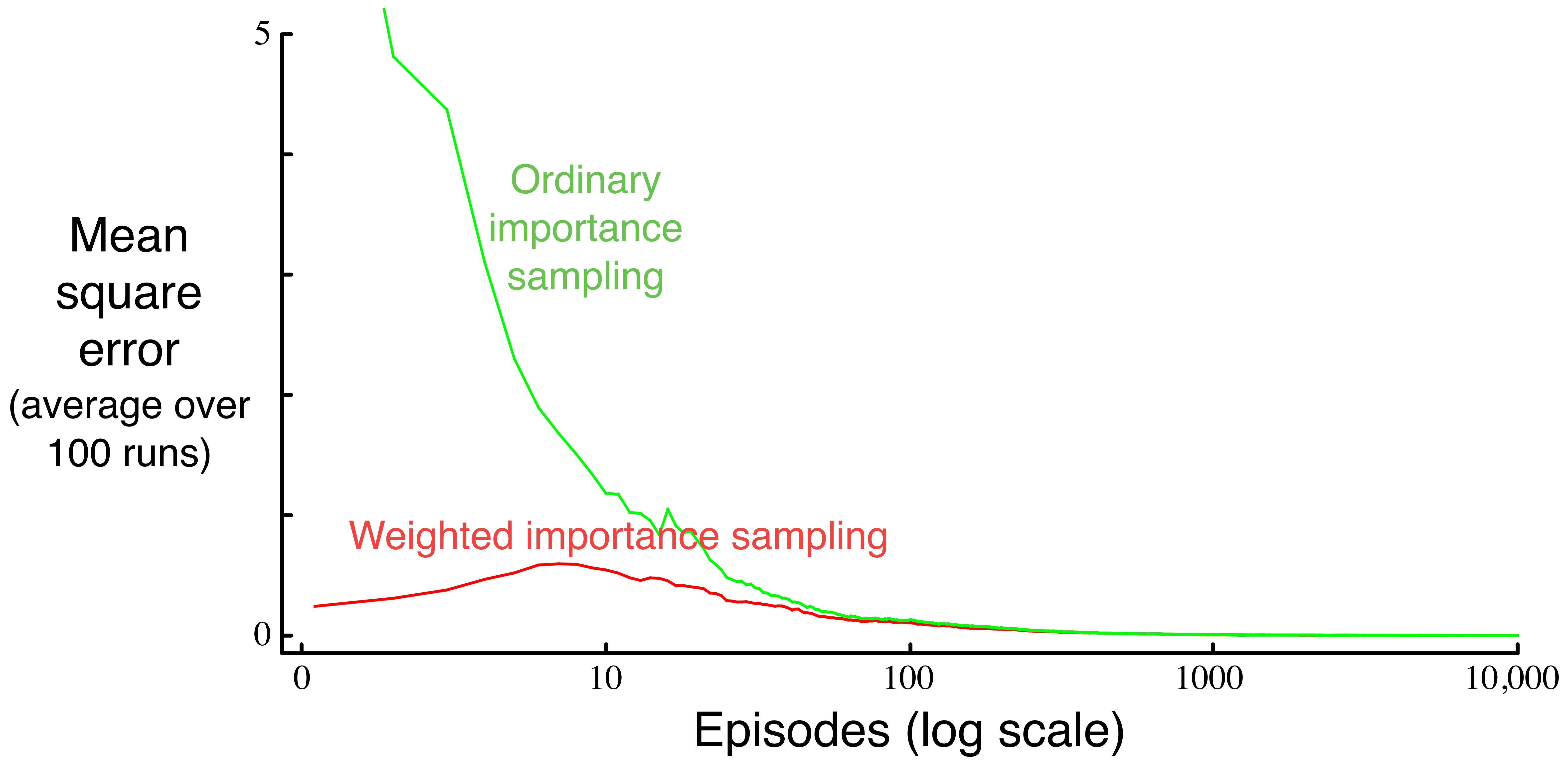


Figure 5.3: Weighted importance sampling produces lower error estimates of the value of a single blackjack state from off-policy episodes. ■

(Image: Sutton & Barto, 2018)

Off-Policy Monte Carlo Prediction

Off-policy MC prediction (policy evaluation) for estimating $Q \approx q_\pi$

Input: an arbitrary target policy π
Initialize, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$:
  $Q(s, a) \in \mathbb{R}$ (arbitrarily)
  $C(s, a) \leftarrow 0$

Loop forever (for each episode):
  $b \leftarrow$ any policy with coverage of π
  Generate an episode following b: $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$
  $G \leftarrow 0$
  $W \leftarrow 1$
  Loop for each step of episode, $t = T-1, T-2, \ldots, 0$, while $W \neq 0$:
    $G \leftarrow \gamma G + R_{t+1}$
    $C(S_t, A_t) \leftarrow C(S_t, A_t) + W$
    $Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \frac{W}{C(S_t, A_t)} [G - Q(S_t, A_t)]$
    $W \leftarrow W \frac{\pi(A_t|S_t)}{b(A_t|S_t)}$

Off-Policy Monte Carlo Control

Off-policy MC control, for estimating $\pi \approx \pi_*$

Initialize, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$:
  $Q(s, a) \in \mathbb{R}$ (arbitrarily)
  $C(s, a) \leftarrow 0$
  $\pi(s) \leftarrow \arg\max_a Q(s, a)$ (with ties broken consistently)

Loop forever (for each episode):
  $b \leftarrow$ any soft policy
  Generate an episode using b: $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$
  $G \leftarrow 0$
  $W \leftarrow 1$
  Loop for each step of episode, $t = T-1, T-2, \ldots, 0$:
    $G \leftarrow \gamma G + R_{t+1}$
    $C(S_t, A_t) \leftarrow C(S_t, A_t) + W$
    $Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \frac{W}{C(S_t, A_t)} [G - Q(S_t, A_t)]$
    $\pi(S_t) \leftarrow \arg\max_a Q(S_t, a)$ (with ties broken consistently)
    If $A_t \neq \pi(S_t)$ then exit inner Loop (proceed to next episode)
    $W \leftarrow W \frac{1}{b(A_t|S_t)}$

Off-Policy Monte Carlo Control

Off-policy MC control, for estimating $\pi \approx \pi_*$

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

  $Q(s,a) \in \mathbb{R}$ (arbitrarily)

  $C(s,a) \leftarrow 0$

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

Loop forever (for each episode):

  $b \leftarrow$ any soft policy

  Generate an episode using b:

  $G \leftarrow 0$

  $W \leftarrow 1$

  Loop for each step of episode, $t = T-1, T-2, \ldots, 0$:

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

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

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

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

    If $A_t \neq \pi(S_t)$ then exit inner Loop (proceed to next episode)

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

$$Q_n = \frac{\sum_{i=1}^{n} W_i G_i}{\sum_{i=1}^{n} W_i} = \frac{\sum_{i=1}^{n} W_i G_i}{C - W}$$

$$Q_{n+1} = \frac{\sum_{i=1}^{n+1} W_i G_i}{\sum_{i=1}^{n+1} W_i} = \frac{(C - W)Q_n + WG}{C}$$

$$= \frac{C}{C} Q_n - \frac{W}{C} Q_n + \frac{W}{C} G = Q_n + \frac{W}{C} [G - Q_n]$$

Questions:

1. Will this procedure converge to the optimal policy π^*?

2. Why do we break when $A_t \neq \pi(S_t)$?

3. Why do the weights W not involve $\pi(A_t \mid S_t)$?

Summary

- Estimating action values requires either **exploring starts** or a **soft policy** (e.g., ϵ-greedy)
- **Off-policy learning** is the estimation of value functions for a **target policy** based on episodes generated by a different **behaviour policy**
  - **Importance sampling** is one way to perform off-policy learning
  - **Weighted** importance sampling has lower **variance** than **ordinary** importance sampling
- **Off-policy control** is learning the **optimal policy** (target policy) using episodes from a **behaviour policy**