Monte Carlo Methods

Reinforcement Learning
Chapter 5

Dev3B 권요한

Contents

Introduction

❖ Monte Carlo Methods

: ways of solving the reinforcement learning problem based on averaging sample returns from *experience*. -> empirical return (cf. expected return)

- Advantage : No need of prior knowledge of the environment's dynamics

-> but infeasible to obtain the probability distributions in explicit form

- Assumption : *Experience* is divided into *episodes*, and that all *episodes* eventually terminate no matter what actions are selected.

- Whereas we computed value functions from knowledge of the MDP in DP, here we learn value functions from sample returns with the MDP.

5.1 Monte Carlo Prediction

First-visit MC prediction

- *Visit* : Each occurrence of a state in an episode
- *First-visit MC method* : estimates $v_\pi(s)$ as the average of the returns following first visits to *s*
- *Every-visit MC method* : estimates $v_\pi(s)$ as the averages of the returns following all visits to *s*

- In this chapter, we focus on the first-visit MC method. Every-visit MC method extends more naturally to function approximation and eligibility traces, as discussed in Chapter 9 and 12.

5.1 Monte Carlo Prediction

❖ First-visit MC prediction

- In this case each return is an *iid* distributed estimate of $v_\pi(s)$ with finite variance.
- By the law of large numbers ; $s \to \infty, V(s) \to v_\pi(s), sd \to 1/\sqrt{n}$

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 $S_0, S_1, \ldots, S_{t-1}$:

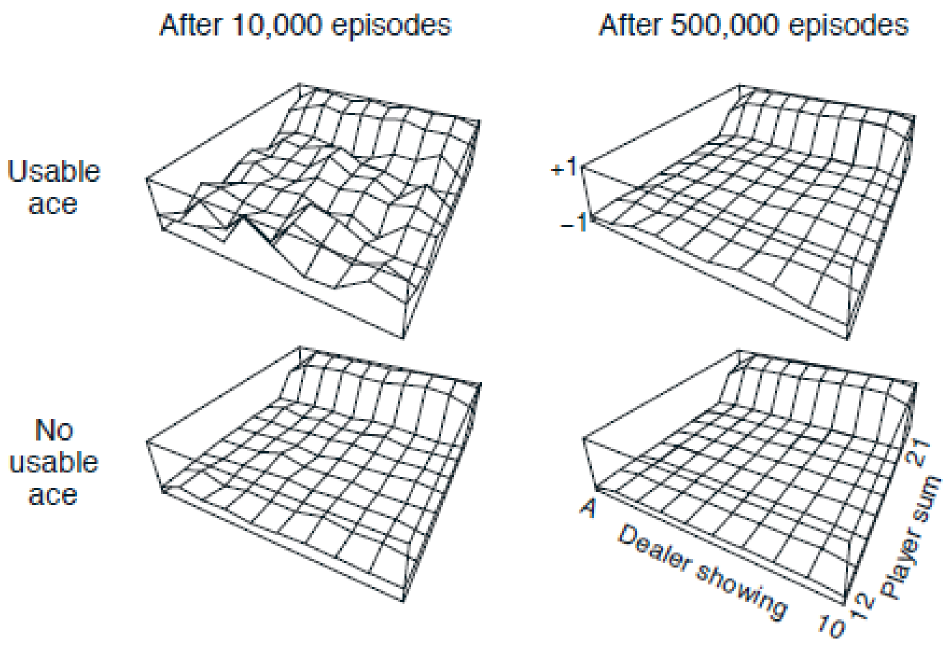
            Append G to $Returns(S_t)$

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

5.1 Monte Carlo Prediction

❖ Example 5.1: Blackjack

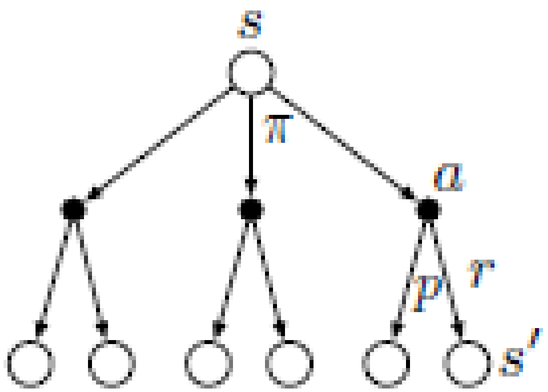
- Episode : Each game of blackjack
- Rewards : +1 for win, -1 for lose, 0 for draw
- Discount rate $\gamma = 1$
- Actions : hit or stick
- States : player's card and dealer's showing card
- Usable ace : counted as 11 without going bust
- Policy : sticks if the player's sum is 20 or 21, and otherwise hits.



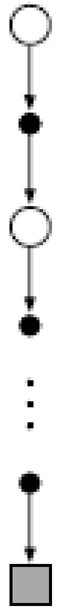
5.1 Monte Carlo Prediction

❖ Backup diagram

DP



MC



Less computation
High variance
Low bias(randomness)

5.2 Monte Carlo Estimation of Action Value

❖ Policy evaluation problem

- If a model is not available, then it is particularly useful to estimate action values rather than state values.

$$\begin{aligned}
\pi'(s) &\doteq \arg\max_a q_\pi(s,a) \\
&= \arg\max_a \mathbb{E}[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s, A_t = a] \qquad (4.9) \\
&= \arg\max_a \sum_{s',r} p(s',r|s,a)\Big[r + \gamma v_\pi(s')\Big],
\end{aligned}$$

- In following a deterministic policy, one will observe returns only for one of the actions from each state because MC is episodic.

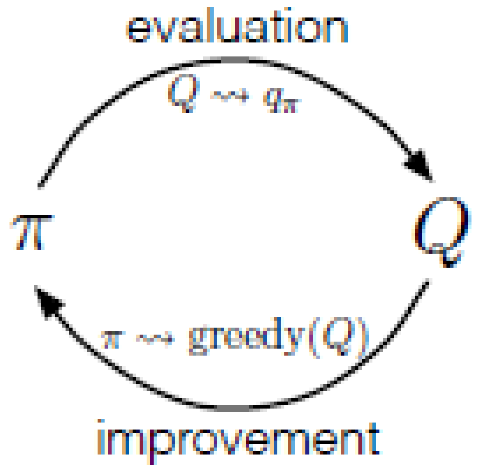
-> problem of *maintaining exploration*

- *Exploring starts* : the episodes start in a state-action pair, and that every pair has a nonzero probability of being selected as the start. -> unrealistic assumption

5.3 Monte Carlo Control

❖ Policy iteration

- $\pi_0 \xrightarrow{\text{E}} q_{\pi_0} \xrightarrow{\text{I}} \pi_1 \xrightarrow{\text{E}} q_{\pi_1} \xrightarrow{\text{I}} \pi_2 \xrightarrow{\text{E}} \cdots \xrightarrow{\text{I}} \pi_* \xrightarrow{\text{E}} q_*,$
- $$\begin{aligned} q_{\pi_k}(s, \pi_{k+1}(s)) &= q_{\pi_k}(s, \arg\max_a q_{\pi_k}(s,a)) \\ &= \max_a q_{\pi_k}(s,a) \\ &\geq q_{\pi_k}(s, \pi_k(s)) \\ &\geq v_{\pi_k}(s). \end{aligned}$$



- Reducing steps and episodes to be useful in practice, we alternate between improvement and evaluation steps for single states. (cf. value iteration)

5.3 Monte Carlo Control

❖ 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 $Returns(S_t, A_t)$
      $Q(S_t, A_t) \leftarrow \text{average}(Returns(S_t, A_t))$
      $\pi(S_t) \leftarrow \text{argmax}_a\, Q(S_t, a)$

- Convergence to this optimal fixed point seems inevitable as the changes to the action-value function decrease over time, but has not yet been formally proved.

5.4 Monte Carlo Control without Exploring Starts

❖ On-policy method

- *On-policy* methods : evaluate or improve a policy that is used to make decisions
- *Off-policy* methods : evaluate or improve a policy different from that used to generate the data
- $\varepsilon - greedy$ policy

- Minimal probability of selection : $\frac{\varepsilon}{|A(s)|}$

- Probability of greedy action : $1 - \varepsilon + \frac{\varepsilon}{|A(s)|}$

$$\begin{aligned}
q_\pi(s, \pi'(s)) &= \sum_a \pi'(a|s) q_\pi(s,a) \\
&= \frac{\varepsilon}{|\mathcal{A}(s)|} \sum_a q_\pi(s,a) + (1-\varepsilon) \max_a q_\pi(s,a) \qquad (5.2) \\
&\geq \frac{\varepsilon}{|\mathcal{A}(s)|} \sum_a q_\pi(s,a) + (1-\varepsilon) \sum_a \frac{\pi(a|s) - \frac{\varepsilon}{|\mathcal{A}(s)|}}{1-\varepsilon} q_\pi(s,a) \\
&= \frac{\varepsilon}{|\mathcal{A}(s)|} \sum_a q_\pi(s,a) - \frac{\varepsilon}{|\mathcal{A}(s)|} \sum_a q_\pi(s,a) + \sum_a \pi(a|s) q_\pi(s,a) \\
&= v_\pi(s).
\end{aligned}$$

5.4 Monte Carlo Control without Exploring Starts

❖ On-policy method

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

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

5.5 Off-policy Prediction via Importance Sampling

❖ Off-policy method

- Dilemma : seek to learn action values conditional on subsequent optimal behavior, but they need to behave non-optimally to explore all actions.

- *Target policy* : the policy being learned
- *Behavior policy* : the policy used to generate behavior

- Off-policy methods are often of greater variance and are slower to converge. On the other hand, off-policy methods are more powerful and general.

5.5 Off-policy Prediction via Importance Sampling

❖ Importance Sampling

- Assumption of *coverage* : $\pi(a|s) > 0$ *implies* $b(a|s) > 0$
- *Importance sampling* : estimating expected values under one distribution given samples from another.
- $$\begin{aligned}\Pr\{A_t, S_{t+1}, A_{t+1}, \ldots, S_T \mid S_t, A_{t:T-1} \sim \pi\} &= \pi(A_t|S_t)p(S_{t+1}|S_t, A_t)\pi(A_{t+1}|S_{t+1}) \cdots p(S_T|S_{T-1}, A_{T-1}) \\ &= \prod_{k=t}^{T-1} \pi(A_k|S_k)p(S_{k+1}|S_k, A_k),\end{aligned}$$
- The importance sampling ratio

$$\rho_{t:T-1} \doteq \frac{\prod_{k=t}^{T-1} \pi(A_k|S_k)p(S_{k+1}|S_k, A_k)}{\prod_{k=t}^{T-1} b(A_k|S_k)p(S_{k+1}|S_k, A_k)} = \prod_{k=t}^{T-1} \frac{\pi(A_k|S_k)}{b(A_k|S_k)}. \tag{5.3}$$

5.5 Off-policy Prediction via Importance Sampling

❖ Importance Sampling

- $\mathbb{E}[G_t|S_t=s] = v_b(s)$
- $\mathbb{E}[\rho_{t:T-1}G_t \mid S_t=s] = v_\pi(s).$ (5.4)

- $T(s)$: The set of all time steps state *s* is visited in (every-visit)
- $T(t)$: The first time of termination following time t
- *Ordinary importance sampling* *weighted importance sampling*

$$V(s) \doteq \frac{\sum_{t\in\mathcal{T}(s)} \rho_{t:T(t)-1}G_t}{|\mathcal{T}(s)|}.$$

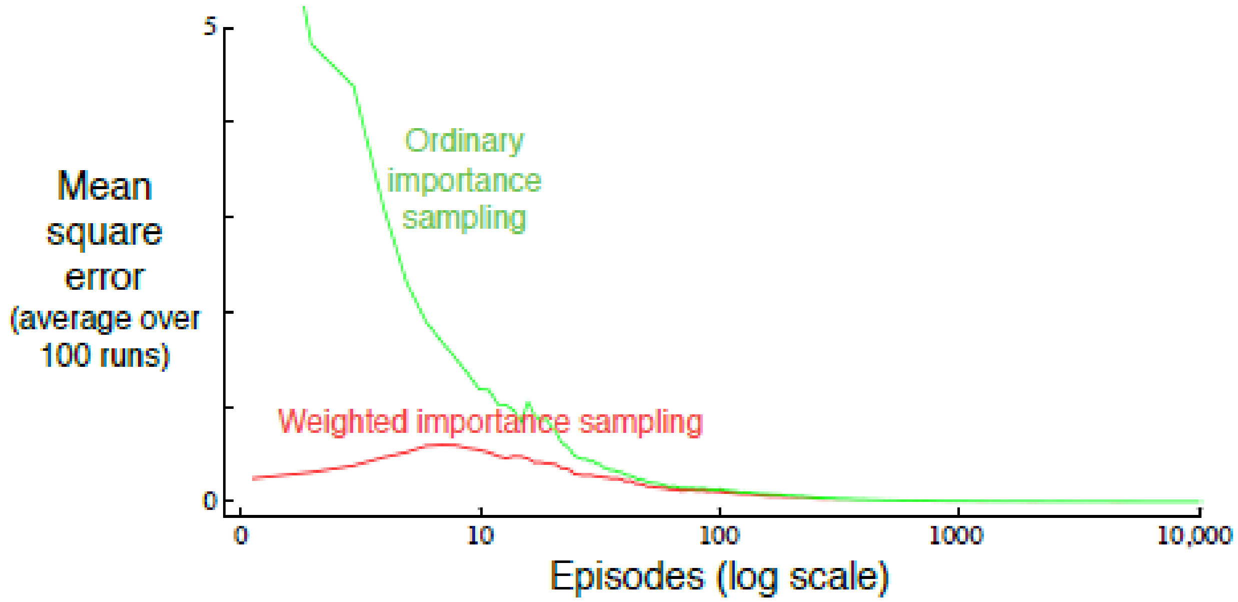
$$V(s) \doteq \frac{\sum_{t\in\mathcal{T}(s)} \rho_{t:T(t)-1}G_t}{\sum_{t\in\mathcal{T}(s)} \rho_{t:T(t)-1}},$$

unbiased bounded variance

5.5 Off-policy Prediction via Importance Sampling

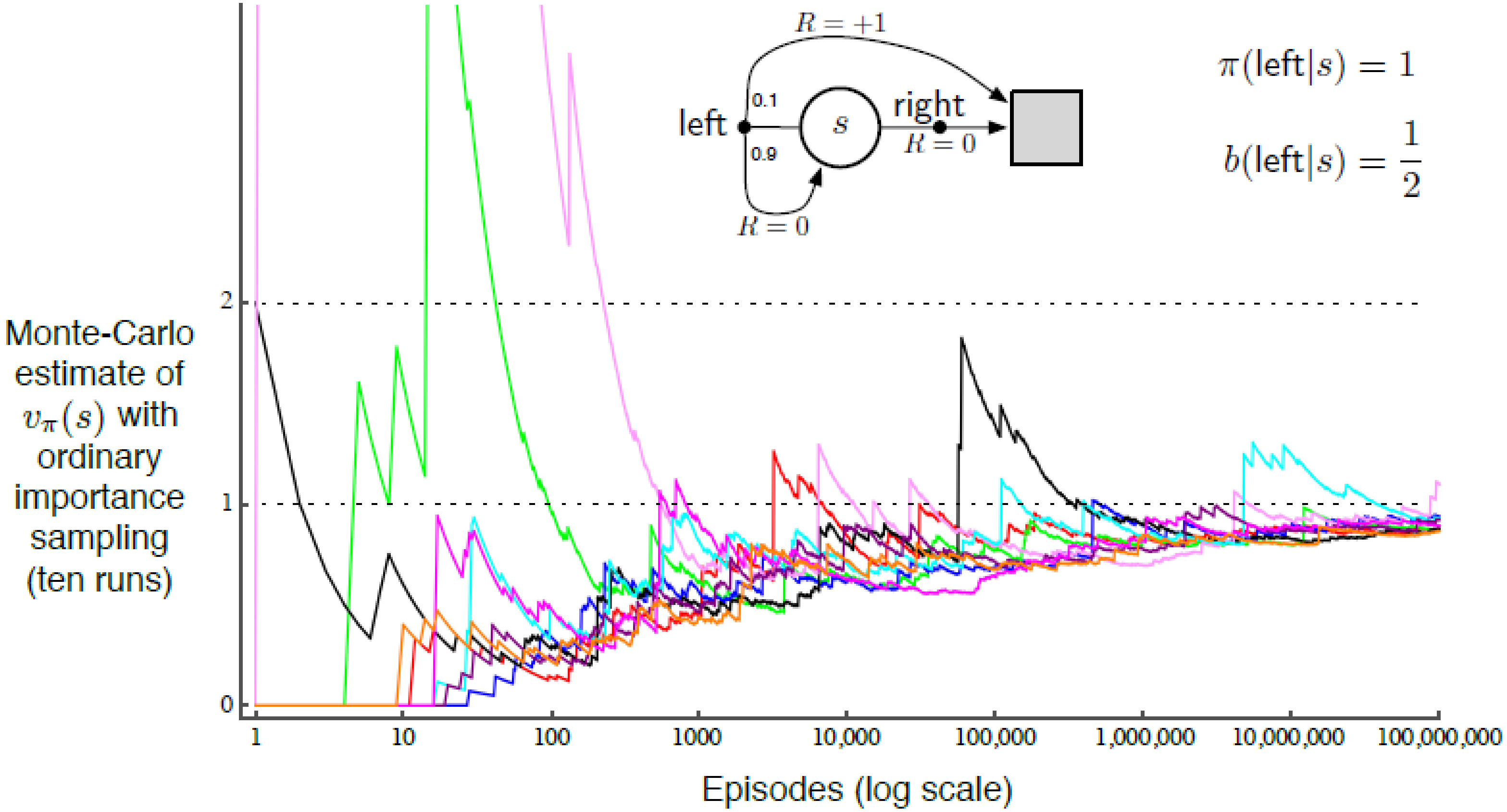
❖ Example 5.4: Off-policy estimation of a Blackjack State Value

- Start state : dealer is showing a deuce, the sum of player's cards is 13, and the player has a usable ace
- Behavior policy : hit or stick with equal probability
- Target policy : stick only on a sum of 20 or 21



5.5 Off-policy Prediction via Importance Sampling

❖ Example 5.5: Infinite Variance



5.6 Incremental Implementation

❖ Incremental methods in weighted importance sampling

- $$V_n \doteq \frac{\sum_{k=1}^{n-1} W_k G_k}{\sum_{k=1}^{n-1} W_k}, \qquad n \geq 2, \tag{5.7}$$

- $$V_{n+1} \doteq V_n + \frac{W_n}{C_n}\Big[G_n - V_n\Big], \qquad n \geq 1, \tag{5.8}$$

and

$$C_{n+1} \doteq C_n + W_{n+1}, \text{ where } C_0 \doteq 0$$

5.6 Incremental Implementation

- Incremental methods in weighted importance sampling

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 $W \neq 0$:
        $G \leftarrow \gamma G + R_{t+1}$
        $C(S_t, A_t) \leftarrow C(S_t, A_t) + W$
        $Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \frac{W}{C(S_t,A_t)} [G - Q(S_t, A_t)]$
        $W \leftarrow W \frac{\pi(A_t|S_t)}{b(A_t|S_t)}$

5.7 Off-policy Monte Carlo Control

❖ Off-policy MC 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)
    $W \leftarrow W \frac{1}{b(A_t|S_t)}$

- Potential problem

learns only from the tails of episodes -> learning could be greatly slow

5.8 Discounting-aware Importance Sampling

❖ Cutting-edge research

- Consider the case where episodes are long and γ is significantly less than 1.

-> enormous variance

- To avoid this large variance, think of discounting as determining a degree of partial termination.

- *Flat partial returns* (*h* is called the *horizon*)

$$\bar{G}_{t:h} \doteq R_{t+1} + R_{t+2} + \cdots + R_h, \qquad 0 \le t < h \le T,$$

- The conventional full return G_t can be viewed

as a sum of flat partial returns.

$$\begin{aligned} G_t &\doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots + \gamma^{T-t-1} R_T \\ &= (1-\gamma) R_{t+1} \\ &\quad + (1-\gamma)\gamma \left(R_{t+1} + R_{t+2}\right) \\ &\quad + (1-\gamma)\gamma^2 \left(R_{t+1} + R_{t+2} + R_{t+3}\right) \\ &\quad\ \vdots \\ &\quad + (1-\gamma)\gamma^{T-t-2} \left(R_{t+1} + R_{t+2} + \cdots + R_{T-1}\right) \\ &\quad + \gamma^{T-t-1} \left(R_{t+1} + R_{t+2} + \cdots + R_T\right) \\ &= (1-\gamma) \sum_{h=t+1}^{T-1} \gamma^{h-t-1} \bar{G}_{t:h} \quad + \quad \gamma^{T-t-1} \bar{G}_{t:T}. \end{aligned}$$

5.8 Discounting-aware Importance Sampling

❖ Discounting-aware importance sampling estimators

- Ordinary importance sampling estimator

$$V(s) \doteq \frac{\sum_{t\in\mathfrak{T}(s)}\left((1-\gamma)\sum_{h=t+1}^{T(t)-1}\gamma^{h-t-1}\rho_{t:h-1}\bar{G}_{t:h} \;+\; \gamma^{T(t)-t-1}\rho_{t:T(t)-1}\bar{G}_{t:T(t)}\right)}{|\mathfrak{T}(s)|}, \quad (5.9)$$

- Weighted importance sampling estimator

$$V(s) \doteq \frac{\sum_{t\in\mathfrak{T}(s)}\left((1-\gamma)\sum_{h=t+1}^{T(t)-1}\gamma^{h-t-1}\rho_{t:h-1}\bar{G}_{t:h} \;+\; \gamma^{T(t)-t-1}\rho_{t:T(t)-1}\bar{G}_{t:T(t)}\right)}{\sum_{t\in\mathfrak{T}(s)}\left((1-\gamma)\sum_{h=t+1}^{T(t)-1}\gamma^{h-t-1}\rho_{t:h-1} \;+\; \gamma^{T(t)-t-1}\rho_{t:T(t)-1}\right)}. \quad (5.10)$$

5.9 Per-decision Importance Sampling

❖ Per-decision importance sampling estimator

- $$\begin{aligned}\rho_{t:T-1}G_t &= \rho_{t:T-1}\left(R_{t+1}+\gamma R_{t+2}+\cdots+\gamma^{T-t-1}R_T\right)\\ &= \rho_{t:T-1}R_{t+1}+\gamma\rho_{t:T-1}R_{t+2}+\cdots+\gamma^{T-t-1}\rho_{t:T-1}R_T. \qquad (5.11)\end{aligned}$$

- $$\rho_{t:T-1}R_{t+1} = \frac{\pi(A_t|S_t)}{b(A_t|S_t)}\frac{\pi(A_{t+1}|S_{t+1})}{b(A_{t+1}|S_{t+1})}\frac{\pi(A_{t+2}|S_{t+2})}{b(A_{t+2}|S_{t+2})}\cdots\frac{\pi(A_{T-1}|S_{T-1})}{b(A_{T-1}|S_{T-1})}R_{t+1}. \quad (5.12)$$

-> only the $\frac{\pi(A_t|S_t)}{b(A_t|S_t)}$ and R_{t+1} are related.

- $$\mathbb{E}\left[\frac{\pi(A_k|S_k)}{b(A_k|S_k)}\right] \doteq \sum_a b(a|S_k)\frac{\pi(a|S_k)}{b(a|S_k)} = \sum_a \pi(a|S_k) = 1. \qquad (5.13)$$

- $$\mathbb{E}[\rho_{t:T-1}R_{t+1}] = \mathbb{E}[\rho_{t:t}R_{t+1}]. \qquad (5.14)$$

5.9 Per-decision Importance Sampling

❖ Per-decision importance sampling estimator

- $\mathbb{E}[\rho_{t:T-1}G_t] = \mathbb{E}\left[\tilde{G}_t\right],$

where

$$\tilde{G}_t = \rho_{t:t}R_{t+1} + \gamma\rho_{t:t+1}R_{t+2} + \gamma^2\rho_{t:t+2}R_{t+3} + \cdots + \gamma^{T-t-1}\rho_{t:T-1}R_T.$$

- Ordinary importance sampling estimator

$$V(s) \doteq \frac{\sum_{t\in\mathfrak{T}(s)} \tilde{G}_t}{|\mathfrak{T}(s)|}, \qquad (5.15)$$

5.10 Summary

❖ Monte Carlo methods' advantages

- The Monte Carlo methods learn value functions and optimal policies from experience in the form of *sample episodes*.
- 1. Learn optimal behavior directly from interaction with the environment, with no model of environment's dynamics.
- 2. Can be used with simulation or sample models.
- 3. Easy and efficient to focus on a small subset of the states.
- 4. Less harmed by violations of the Markov property.

5.10 Summary

❖ Maintaining sufficient exploration in MC control methods

- *Exploring starts* : episodes begin with state-action pairs randomly selected
- In *on-policy* methods : the agent commits to always exploring
- In *off-policy* methods : data generated by a different *behavior policy*

-> based on importance sampling : weighting returns by the ratio of the probabilities of taking the observed actions under two policies

(= transforming expectations from the behavior policy to the target policy)

\- *Ordinary importance sampling*

\- *Weighted importance sampling*

- In the next chapter, we consider methods that learn from experience, like MC methods, but also bootstrap, like DP methods.

Thank you everyone and me!!

- Don’t you have any Questions?