CMPUT 365: Monte Carlo Methods

Rupam Mahmood

Feb 7, 2022

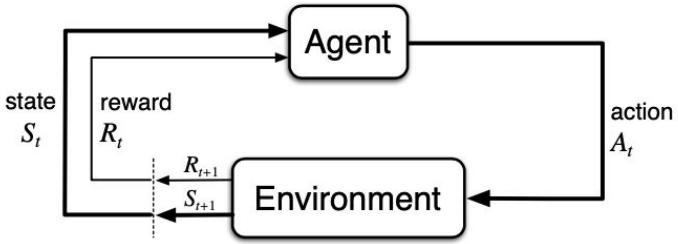
Admin: Coursera course 2 starts from this week

Due dates C2M1

Midterm:

- Exam time: 45 minutes + 5 minutes for uploading
- Syllabus: Coursera course 1 / Chapter 2-4 / up to Dynamic Programming
- The practice midterm is for you to understand the kind of questions to expect
- The answers should be as detailed as the sample answers
- Practicing and making a cheat sheet are way more helpful than hoping that you can figure out the answers by opening the book

Monte Carlo is a sample-based learning method



Benefits:

Vova Selin 对所有人说 下午1:12

V

Since we usually don't know the environment dynamics accurately

David Wang 对所有人说 下午1:14

DW

It saves times. You don't have to update the entire model on every iteration.

...

- Learning from actual experience is striking because it requires no prior knowledge of the environment's dynamics, yet can still attain optimal behavior
- We saw a colab example before of averaging returns
- Monte Carlo (MC) methods are more easily understood in the episodic setting
- Updates can be made only after an episode is done
- Updates can be made for the first visit of a state or every visit of a state

MC 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 $Returns(S_t)$

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

- After each update here, the average V remains an unbiased estimate of the value function
- How do you convert this into an every-visit one?

MC estimation of Action Values

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, $t = T-1, T-2, \ldots, 0$:

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

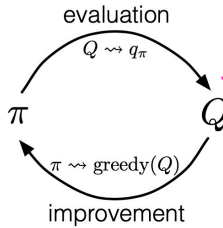
Unless S_t appears in $S_0, S_1, \ldots, S_{t-1}$:

Append G to $Returns(S_t)$

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

- How can we convert this pseudocode for action values?
- Sample-based methods fundamentally rely on exploration and randomness
- Two ways to maintain exploration: exploring start and using stochastic policies

MC control



→ evaluate a policy, and then greedify it.

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

$$\pi(s) \doteq \arg\max_a q(s,a)$$

- MC control must use action-value estimates. Why?
- Let's consider a classical policy iteration version of MC (no model but full evaluation and improvement in turn)
- Policy evaluation is done as before, for example, using exploring start
- Policy improvement is done by making the policy greedy with respect to the current value
- According to the policy improvement theorem, the new policy usually becomes larger or better than the old policy? ⇒ Better, not larger.
- Why this method is impractical?

MC control

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 (explore start.)
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 $Returns(S_t, A_t)$
$Q(S_t, A_t) \leftarrow \text{average}(Returns(S_t, A_t))$
$\pi(S_t) \leftarrow \text{argmax}_a Q(S_t, a)$ (greedily the value estimated.)

- We can give up on completing the evaluation step to avoid sampling infinite number of episodes at each evaluation step
- Stability is achieved only when both the policy and the value function are optimal
- But convergence is yet to be proved

MC control without exploring starts

- We can get rid of exploring start by using a stochastic policy
- We move the policy only to an epsilon-greedy policy. Still works as a GPI

- What's the difference between epsilon-soft and epsilon-greedy policy?
- any epsilon-greedy policy with respect to q pi is an improvement over any epsilon-soft policy pi

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 $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$

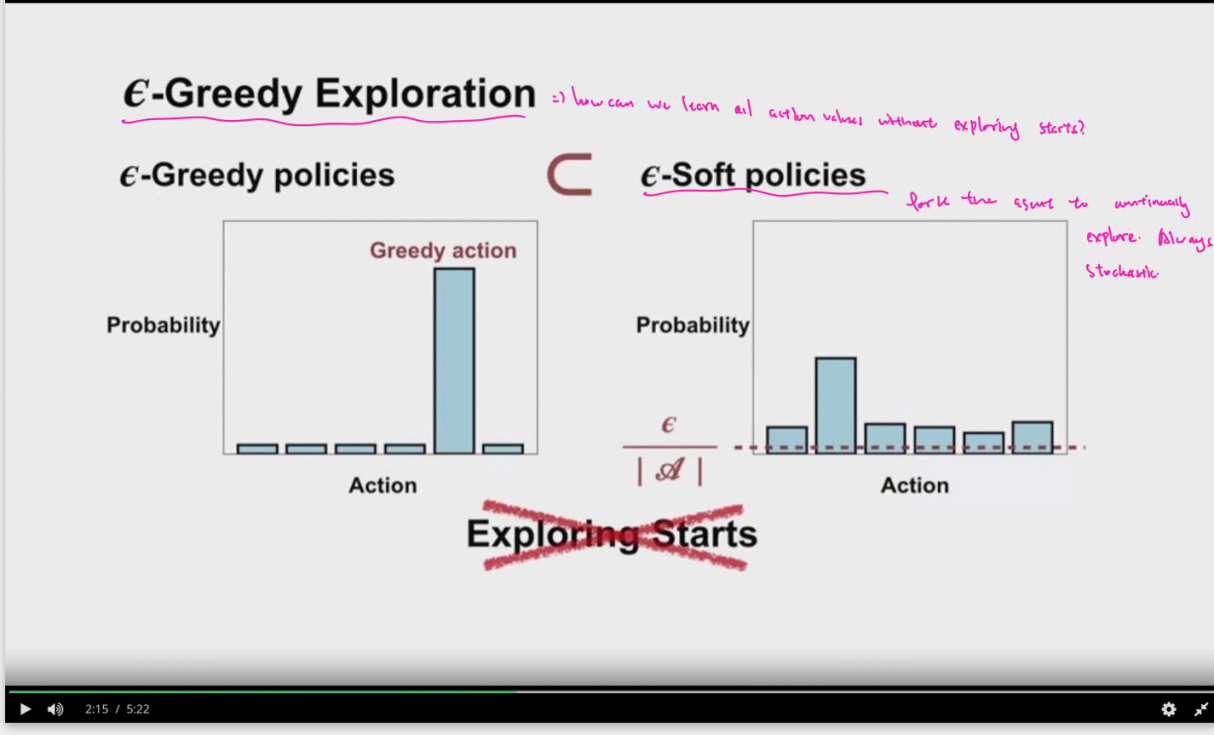
Loop forever (for each episode):
Choose $S_0 \in \mathcal{S}$, $A_0 \in \mathcal{A}(S_0)$ randomly such that all pairs have probability > 0
Generate an episode from S_0, A_0, following π: $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$
$G \leftarrow 0$
Loop for each step of episode, $t = T-1, T-2, \ldots, 0$:
$G \leftarrow \gamma G + R_{t+1}$
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 $Returns(S_t, A_t)$
$Q(S_t, A_t) \leftarrow \text{average}(Returns(S_t, A_t))$
$\pi(S_t) \leftarrow \arg\max_a Q(S_t, a)$

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

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



Epsilon soft policies are always stochastic. Deterministic policy specify a single action to take in each state, stochastic policies instead specify the probability of taking action in each state in epsilon. Soft policies: all actions have a probability of at least Epsilon over the number of actions. They will eventually try all the actions.

History of RL

- Rich S. Sutton
- Andy Barto
- A. Harry Klopf
- Philosophy of science: empiricism, logical positivism, Karl Popper
- Phenomenology: Hubert Dreyfus

[Last slido question]

If return G is obtained from state s and following policy [redacted] b, is G an unbiased estimate of v_pi(s)?

0 1 2

Yes

8%

No

92%

Participants can vote at slido.com with code #X978

Same as asking $\mathbb{E}_b[G_t \mid S_t = s] \overset{?}{\neq} v_\pi(s)$.

$\mathbb{E}_{A_k \sim b}$

different policy from π.

expectation of return.

A random variable G_t doesn't tell much about the probability distribution. don't know how the action is chosen by the agent, ∴ 写了 G 并不能代表 estimate 是明确的，∴ 并非 unbiased!!

An action is drawn from π in this distribution.

[Let's review unbiasedness]

$$Z_n = \frac{\sum_{i=1}^{n} X_i}{n}$$

Z_n: R.V. — Based on the value of n.

X_i: R.V.

$\frac{\sum_{i=1}^{n} X_i}{n}$: sample average estimator.

an unbiased estimate of $E_{X \sim p}[X]$ where $X_i \overset{iid}{\sim} p$?

slido question ...

Same as asking $E_{X_i \sim p}[Z_n] \overset{?}{=} E_{X \sim p}[X]$

- $X_i \sim p$: each of the action is drawn from policy π.
- $X \sim p$: each of the action is drawn from policy P.

[What if data comes from a different distribution?]

Is $Z_n = \frac{\sum_{i=1}^{n} X_i}{n}$ an unbiased estimate of $E_{X \sim p}[X]$ where $X_i \sim d$?

slido question ...

Same as asking $E_{X_i \sim d}[Z_n] \overset{?}{\neq} E_{X \sim p}[X]$.

[unbiasedness of sample average is connected to the unbiasedness of the summand]

$$Z_n = \frac{X_1 + X_2 + \cdots + X_n}{n}$$

X_n ← summand

→ each of this summand are unbiased estimates.

say $\mu \doteq E_{x \sim p}[X]$.

then $E_{X_i \sim p}[X_i] = \mu$ as well

because X, X_i are all i.i.d.

And $E_{X_i \sim p}[X_i] = \mu$

$\Rightarrow E_{X_i \sim p}[Z_n] = \mu$.

$$E_{X_i \sim p}[Z_n] = E_{X_i \sim p}\left[\frac{\sum_{i=1}^{n} X_i}{n}\right]$$

$$= \frac{1}{n}\sum_{i=1}^{n} E_{X_i \sim p}[X_i]$$

expectation of each of this X_i.

$$= \frac{1}{n}\sum_{i=1}^{n} \mu$$

$$= \frac{1}{n} \times n\mu = \mu.$$

$\mu = v_\pi(s)$

$$\text{[Therefore } E_{X_i \sim d}[Z_n] \neq \mu \text{]}$$

each of the samples are drawn from the different distribution d (not p).

$$Z_n = \frac{\sum_{i=1}^{n} X_i}{n} \; ; \; X_i \sim d$$

$$V(s) = \frac{\sum_{t \in \mathcal{T}(s)} G_t}{|\mathcal{T}(s)|} \; ; G_t \sim b$$

$$E_{X_i \sim d}[X_i] \neq \mu = E_{X \sim p}[X]$$

mean of distribution d $\neq$ mean of distribution p.

$\therefore E_b[G_t | S_t = s] \neq V_\pi(s)$. policy π.

estimate return under policy b.

$b \neq \pi$. $\therefore$ 定義不相等 !!

$$E_b[G_t | S_t = s] \neq v_\pi(s)$$

$$E_b[V(s)] \neq v_\pi(s).$$

How about $E_{X_i \sim d}[Z_n'] \stackrel{?}{=} \mu$

$$Z_n' = \frac{\sum_{i=1}^{n} Y_i}{n}, \text{ where}$$

Y_i: sample average y_i (summand)

$$Y_i = \frac{p(X_i)}{d(X_i)} X_i \;\&\; X_i \sim d.$$

- $p(X_i)$: probability (density) of X_i under a distribution.
- $d(X_i)$: probability (density) of X_i under another distribution.
- $\frac{p(X_i)}{d(X_i)}$: importance sampling ratio
- X_i: summand.

Think in terms of summands of Z_n'.

$$E_{X_i \sim d}[Y_i] = \mu \Rightarrow$$

$$E_{X_i \sim d}[Z_n'] = \mu.$$

Y_i: unbiased estimate of μ.

$$E_{X_i \sim d}[Y_i] = E_{X_i \sim d}\left[\frac{p(X_i)}{d(X_i)} X_i\right]$$

$$= \sum \frac{p(x)}{d(x)} x \, d(x)$$

$d(x)$: probability under the base R.V.

$$= \sum p(x) x = E_{X \sim p}[X] = \mu.$$

assumption of coverage

$p(x) > 0 \Rightarrow d(x) > 0.$

- $p(x) > 0$: when ever $p(x)$ is non-zero
- $d(x) > 0$: then $d(x)$ is non-zero.

$$G_t \frac{p(G_t|\pi)}{p(G_t|b)}$$

π → wants to predict π.

b → drawn from b.

Unbiased and consistent estimation

Say $X_i \sim p$ is an iid random variable

The sample average $Z_n = \dfrac{\sum_{i=1}^{n} X_i}{n}$ is an estimate of $E_{X\sim p}[X] = \sum_x xp(x)$

So is X_i

Then we have $E_{X_i\sim p}[Z_n] = E_{X\sim p}[X]$; unbiasedness of Z_n

Sample average

And we have $P\left(\lim_{n\to\infty} Z_n = E_{X\sim p}[X]\right) = 1 \iff Z_n \overset{a.s.}{\to} E_{X\sim p}[X]$; consistency of Z_n

as $n \to \infty$, this estimate is still random.

going closer and closer to the thing we want.

On the other hand, we have $E_{X_i\sim p}[X_i] = E_{X\sim p}[X]$, but not $X_i \overset{a.s.}{\to} E_{X\sim p}[X]$

When samples are from a different distribution …

Say $X_i \sim d$ is an iid random variable (note the difference in distribution)

Let's call d the data distribution, and p the target distribution

The sample average $Z_n = \frac{\sum_{i=1}^{n} X_i}{n}$ is a *bad* estimate of $E_{X \sim p}[X] = \sum_x xp(x)$

Because now we have $E_{X_i \sim d}[Z_n] = E_{X \sim d}[X] \neq E_{X \sim p}[X]$

And $Z_n \overset{a.s.}{\rightarrow} E_{X \sim d}[X] \neq E_{X \sim p}[X]$

When samples are from a different distribution …

Obviously, $X_i \sim d$ **is a worse estimate of** $E_{X\sim p}[X]$

How about $Y_i = \frac{p(X_i)}{d(X_i)} X_i$**, where** $X_i \sim d$**?**

If d **provides adequate coverage of** $p : p(x) > 0$ **implies** $d(x) > 0$,

$$E_{X_i \sim d}\left[Y_i\right] = E_{X_i \sim d}\left[\frac{p(X_i)}{d(X_i)} X_i\right] = \sum_x \frac{p(x)}{d(x)} x d(x)$$

$$= \sum_x x p(x) = E_{X\sim p}[X]$$

When samples are from a different distribution, we can use importance sampling correction

$\frac{p(X_i)}{d(X_i)}$ **is known as the importance sampling ratio**

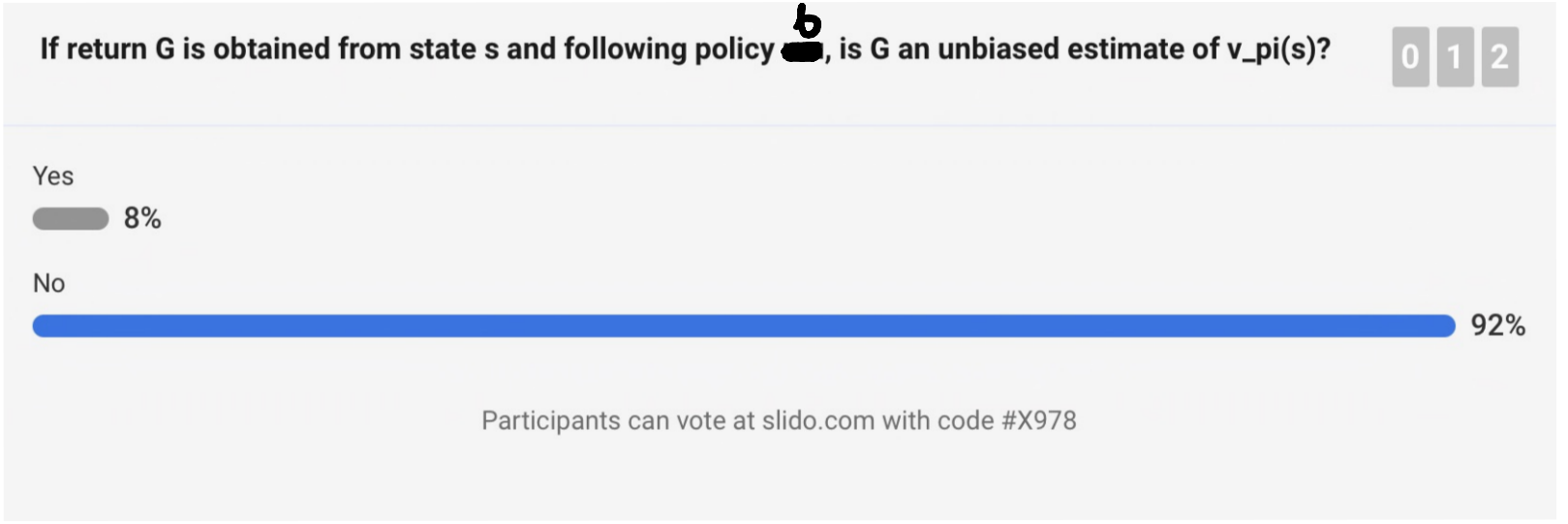
It can be used to correct the discrepancy between target and data distributions

The following importance sampling estimator is an unbiased and consistent

estimator of $E_{X \sim p}[X]$

$$Z_n^{'} = \frac{\sum_{i=1}^{n} Y_i}{n}, \text{ where } Y_i = \frac{p(X_i)}{d(X_i)} X_i \text{ and } X_i \sim d$$

[Last slido question]



Same as asking $E_b[G_t | S_t = s] \overset{?}{=} v_\pi(s)$.

$E_b[G_t | S_t = s] \neq v_\pi(s)$ because $E_b[G_t | S_t = s] = v_b(s)$

and $b \neq \pi$ in general.

Importance sampling for off-policy prediction

We want to estimate v_π whereas samples are from a different policy $b \neq \pi$

We call b the behavior policy, and π the target policy

Then the importance sampling ratio for a trajectory corresponding to return G_t is

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

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

unbiased estimate.

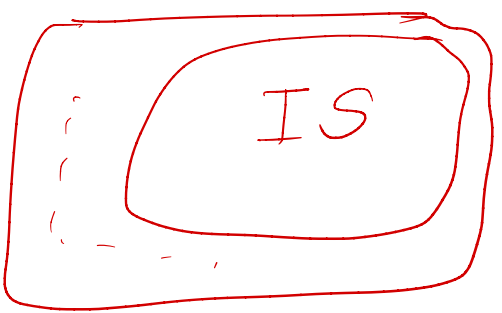
Importance sampling estimator for off-policy prediction: $V(s) \doteq \dfrac{\sum_{t \in \mathcal{T}(s)} \rho_{t:T(t)-1} G_t}{|\mathcal{T}(s)|}$

Here $A_k \sim b$.

Hence, $E_b[V(s)] = v_\pi(s)$.

$V(s)$ is an importance sampling estimator and also an off-policy estimator.

$E_{A \sim b}[\rho_{t:T(t)-1} G_t] = v_\pi(s)$.



Importance sampling for off-policy prediction

Sample average estimator for on-policy prediction: $V(s) \doteq \frac{\sum_{t \in \mathcal{T}(s)} G_t}{|\mathcal{T}(s)|}$

sample average

$\mathcal{T}(s)$ **contains all time steps in which state** s **is visited**

G_t **denotes the return after** t **up through** $T(t)$

$T(t)$ **denotes the first time of termination after** t

Importance sampling estimator for off-policy prediction: $V(s) \doteq \frac{\sum_{t \in \mathcal{T}(s)} \rho_{t:T(t)-1} G_t}{|\mathcal{T}(s)|}$

Harsh Shah对所有人说 下午1:52

HS

can a decision made based on the feedback loop sometime deviates away from your ultimate goal because here you are now focused on the immediate reward? ⇒ the algorithm is decided for focus on the 长期收益 (折扣 γ (discount rate), 可以设为1).

Worksheet 6: Monte Carlo

CMPUT 397
February 8, 2022

1. (*Exercise 5.4 S&B*) The pseudocode for *Monte Carlo ES* is inefficient because, for each state-action pair, it maintains a list of all returns and repeatedly calculates their mean. How can we modify the algorithm to have incremental updates for each state-action pair?

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