RL Excercises Chapter 5

April 28, 2025

5 Exercises

5.1 Question

Consider the diagrams on the right in Figure 5.1. Why does the estimated value function jump up for the last two rows in the rear? Why does it drop off for the whole last row on the left? Why are the frontmost values higher in the upper diagrams than in the lower?

Answer

- Policy is to hit unless $S \geq 20$. So you run a rik of going bust if you have 12-19, but you most likely win when you stick on 20 or 21
- Drops off because dealer has a usable ace
- Frontmost row is higher in the upper diagram because you have one chance more in the sense that if you bust one time, you can still use the Ace.

5.2 Question

Suppose every-visit MC was used instead of first-visit MC on the blackjack task. Would you expect the results to be very different? Why or why not?

Answer

The results would be the same because a state can be visited only once.

5.3 Question

What is the backup diagram for Monte Carlo estimation of q pi ?

Answer

The backup diagram for MC is given at page 95. An empty circle and a black filled circle represent a state action pair.

5.4 Question

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. It would be more efficient to use techniques similar to those explained in Section 2.4 to maintain just the mean and a count (for each state–action pair) and update them incrementally. Describe how the pseudocode would be altered to achieve this.

Answer

In section 2.4 average return is calculated iteratively using only current average and number of occurrences.

Average return for each state-action pair can be calculated similarly.

$$Q(S,A) = \frac{Q(S,A) \times (N(S,A)-1) + G}{N(S,A)} = Q(S,A) + (G - Q(S,A))/N(S,A) \tag{1}$$

5.5 Question

Consider an MDP with a single nonterminal state and a single action that transitions back to the nonterminal state with probability p and transitions to the terminal state with probability 1-p. Let the reward be +1 on all transitions, and let $\gamma = 1$. Suppose you observe one episode that lasts 10 steps, with a return of 10. What are the first-visit and every-visit estimators of the value of the nonterminal state?

Answer

There is no separate behaviour policy. Thus importing sampling ratio is 1.

First visit case:
$v(s) = 1 * 10/1 = 10$
Every visit case:
At the end of the algorithm $Returns(s) = [1,2,3,4,5,6,7,8,9,10]$, and $v(s) = \frac{1*10+1*9+1*8+1*7+1*6+1*5+1*4+1*3+1*2+1*1}{10} = 55/10 = 5.5$
Results are similar if weighted estimator is used.

5.6 Question

What is the equation analogous to (5.6) for action values Q(s, a) instead of state values V (s), again given returns generated using b?

Answer

$$Q(s,a) = \frac{\sum_{t \in \mathcal{T}(s,a)} \rho_{t:(T(t)-1)} G_t}{\sum_{t \in \mathcal{T}(s,a)} \rho_{t:(T(t)-1)}} \tag{2}$$

5.7 Question

In learning curves such as those shown in Figure 5.3 error generally decreases with training, as indeed happened for the ordinary importance-sampling method. But for the weighted importance-sampling method error first increased and then decreased. Why do you think this happened?

Answer

Weightet importance-sampling initially produces biased estimations. As the number of samples increases the bias disappears. It is explained when *weighted importance-sampling* was introduced.

5.8 Question

The results with Example 5.5 and shown in Figure 5.4 used a first-visit MC method. Suppose that instead an every-visit MC method was used on the same problem. Would the variance of the estimator still be infinite? Why or why not?

Answer

No lo revisé con cuidado.

The only difference would be the added 1/k term to the sum. The variance would still be infinite.

5.9 Question

Modify the algorithm for first-visit MC policy evaluation (Section 5.1) to use the incremental implementation for sample averages described in Section 2.4.

Answer

Update rule should be replaced with the following expression.

$$V(S_t) = V(S_t) + (G - V(S_t))/|\tau(S_t)| \tag{3}$$

5.10 Question

Derive the weighted-average update rule (5.8) from (5.7). Follow the pattern of the derivation of the unweighted rule (2.3).

Answer

An important alternative is weighted importance sampling, which uses a weighted average, defined as:

$$V(s) = \frac{\sum_{t\in\tau(s)} \rho_{t:T(t)-1} G_t}{\sum_{t\in\tau(s)} \rho_{t:T(t)-1}} \tag{4}$$

Sequence of returns $G_1, G_2, \ldots, G_{n-1}$, all starting in the same state and each with a corresponding random weight W_i (e.g., $W_i = \rho_{t_i:T(t_i)-1}$).

$$V_n = \frac{\sum_{k=1}^{n-1} W_k G_k}{\sum_{k=1}^{n-1} W_k} \tag{5}$$

We have C_n defined as :

$$C_n = \sum_{k=1}^{n} W_k \tag{6}$$

The update rule would follow:

$$V_{n+1} = \frac{\sum_{k=1}^{n} W_k G_k}{\sum_{k=1}^{n} W_k} = \frac{1}{\sum_{k=1}^{n} W_k} \sum_{k=1}^{n} W_k G_k = \frac{1}{C_n}(W_n G_n + \sum_{k=1}^{n-1} W_k G_k) \tag{7}$$

$$V_{n+1} = \frac{1}{C_n}(W_n G_n + \frac{\sum_{k=1}^{n-1} W_k}{\sum_{k=1}^{n-1} W_k} \sum_{k=1}^{n-1} W_k G_k) = \frac{1}{C_n}(W_n G_n + V_n \sum_{k=1}^{n-1} W_k) \tag{8}$$

$$V_{n+1} = \frac{1}{C_n}(W_n G_n + V_n(\sum_{k=1}^{n} W_k - W_n)) = \frac{1}{C_n}(W_n G_n + V_n(C_n - W_n)) \tag{9}$$

$$V_{n+1} = \frac{1}{C_n}(W_n G_n + V_n C_n - V_n W_n) = \frac{1}{C_n}(V_n C_n + W_n(G_n - V_n)) \quad (10)$$

$$V_{n+1} = V_n + \frac{W_n(G_n - V_n)}{C_n} \quad (11)$$

5.11 Question

In the boxed algorithm for off-policy MC control, you may have been expecting the W update to have involved the importance-sampling ratio $\frac{\pi(A_t|S_t)}{b(A_t|S_t)}$, but instead it involves $\frac{1}{b(A_t|S_t)}$. Why is this nevertheless correct?

Answer

π is a deterministic policy (e.g. argmax is used.). π returns 1 for the greedy action and 0 otherwise. The importance-sampling ratio will become 0 after policy π and b diverges. As a matter of the fact, algorithm will quit the actions loop once $A_t \neq \pi(S_t)$.

In this setting it is OK to use 1 instead of $\pi(A_t|S_t)$ which already equals to 1.

5.12 Question

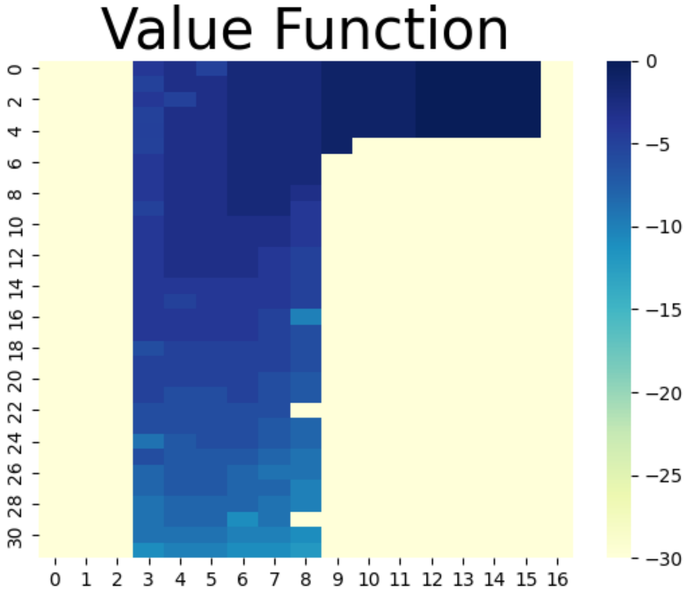
Racetrack (programming)

Consider driving a race car around a turn like those shown in Figure 5.5. You want to go as fast as possible, but not so fast as to run off the track.

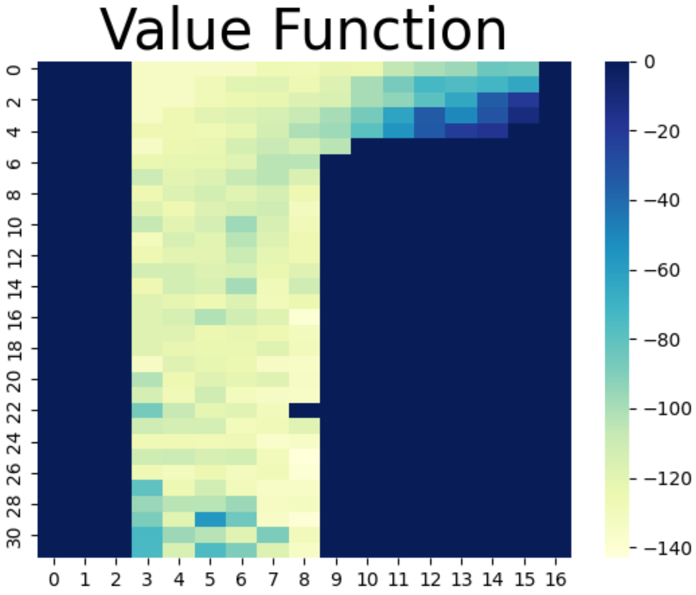
Apply a Monte Carlo control method to this task to compute the optimal policy from each starting state. Exhibit several trajectories following the optimal policy (but turn the noise o for these trajectories).

Answer

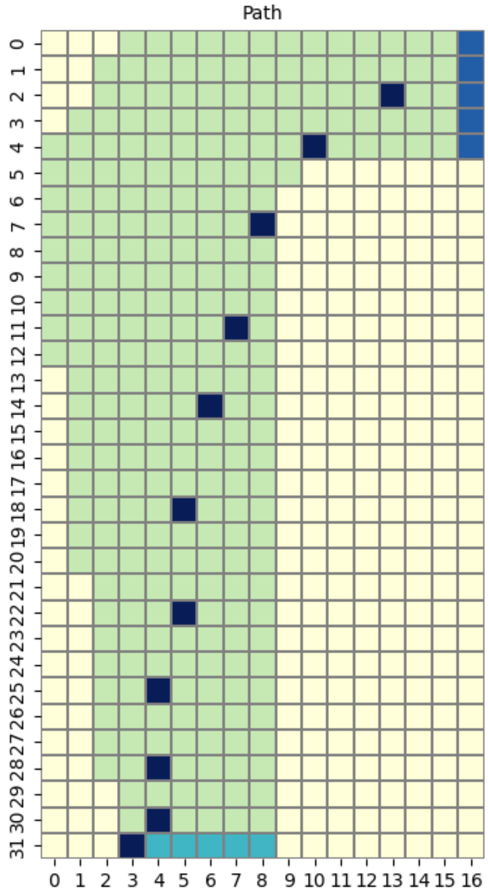
Grid 1 (env2)



Maximum values over speeds.

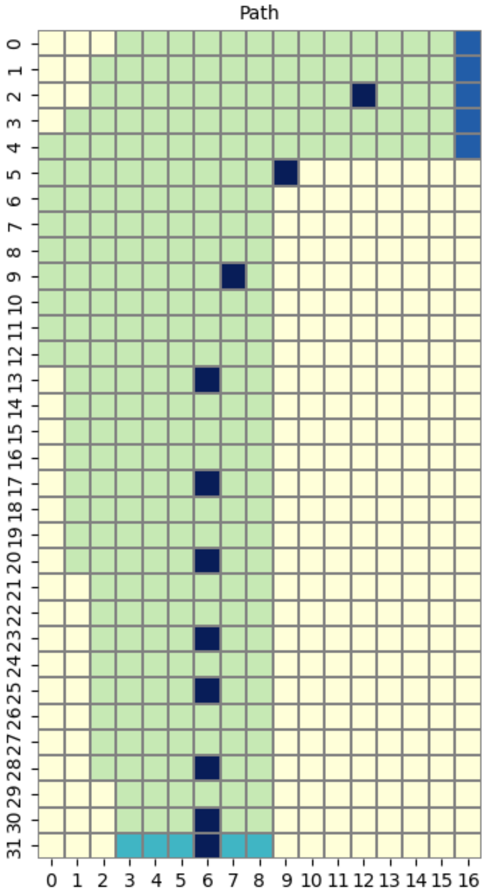


Average values over speeds.

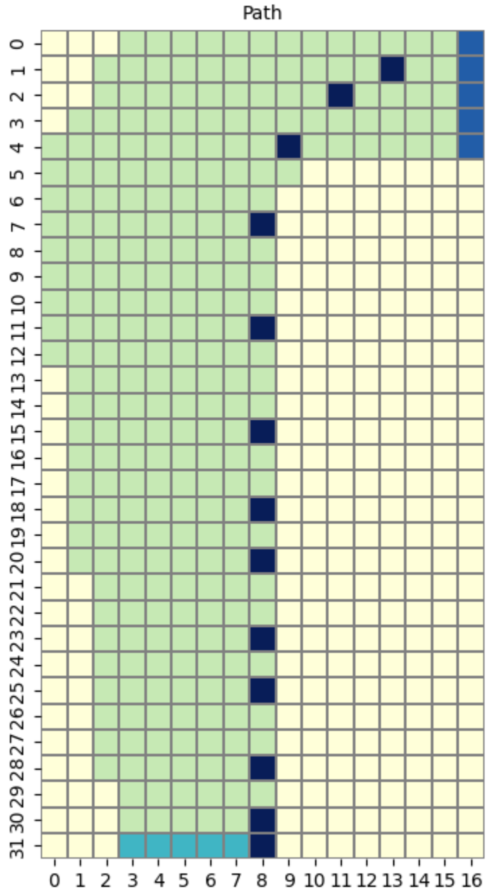


Path

Demo path.

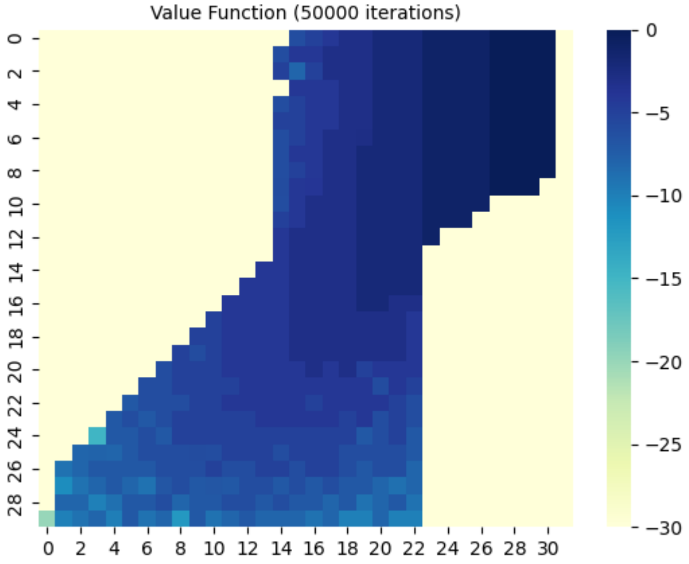

Path
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16

Demo path.

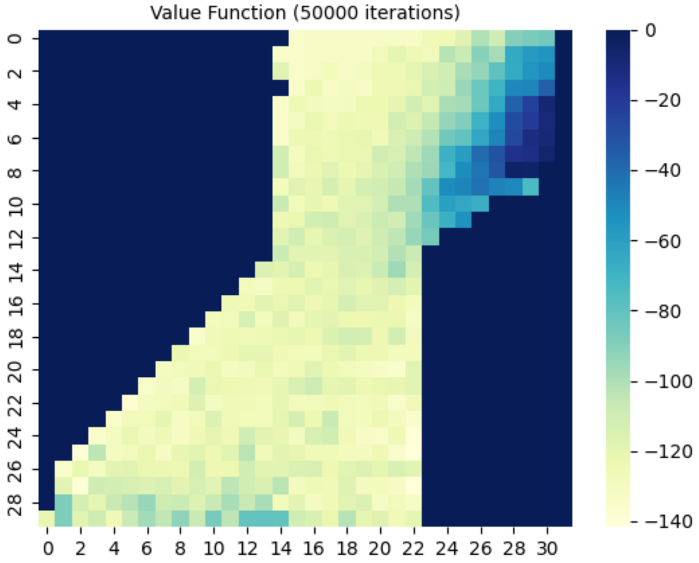


Demo path.
Average values over speeds.

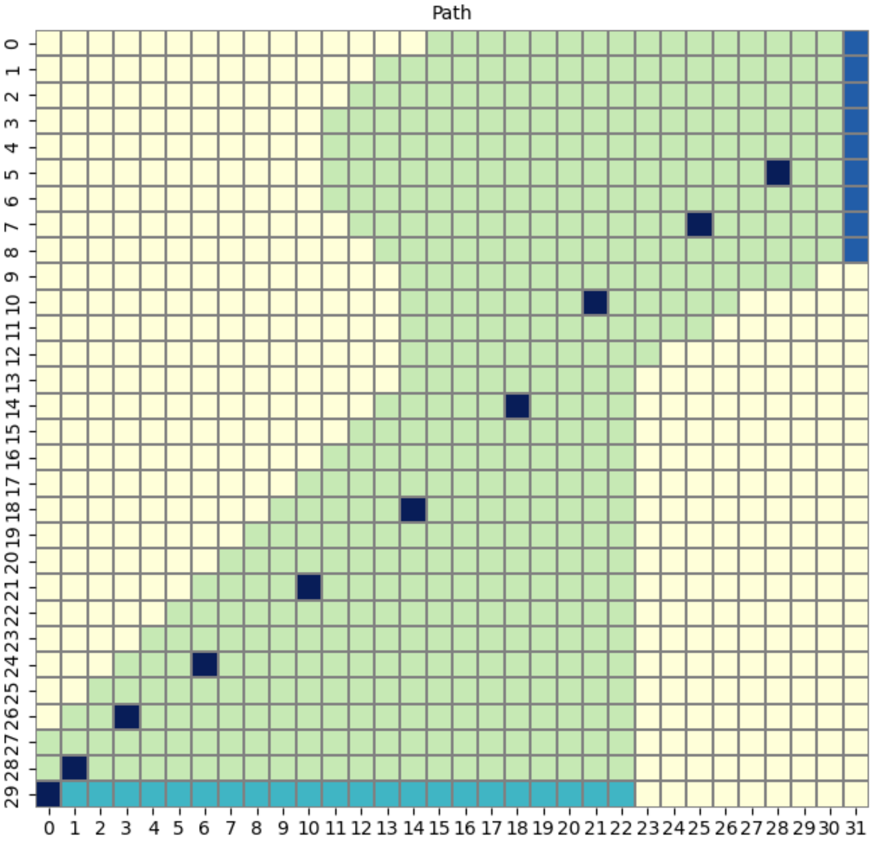
5.12.1 Grid 2 (env3)



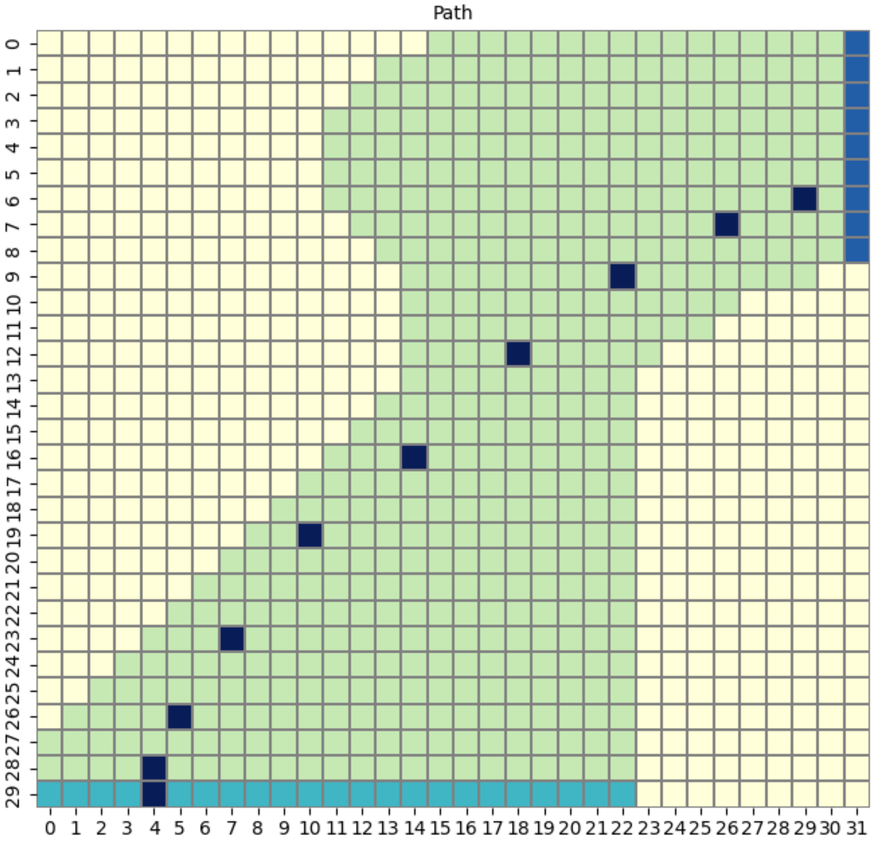
Maximum values over speeds.



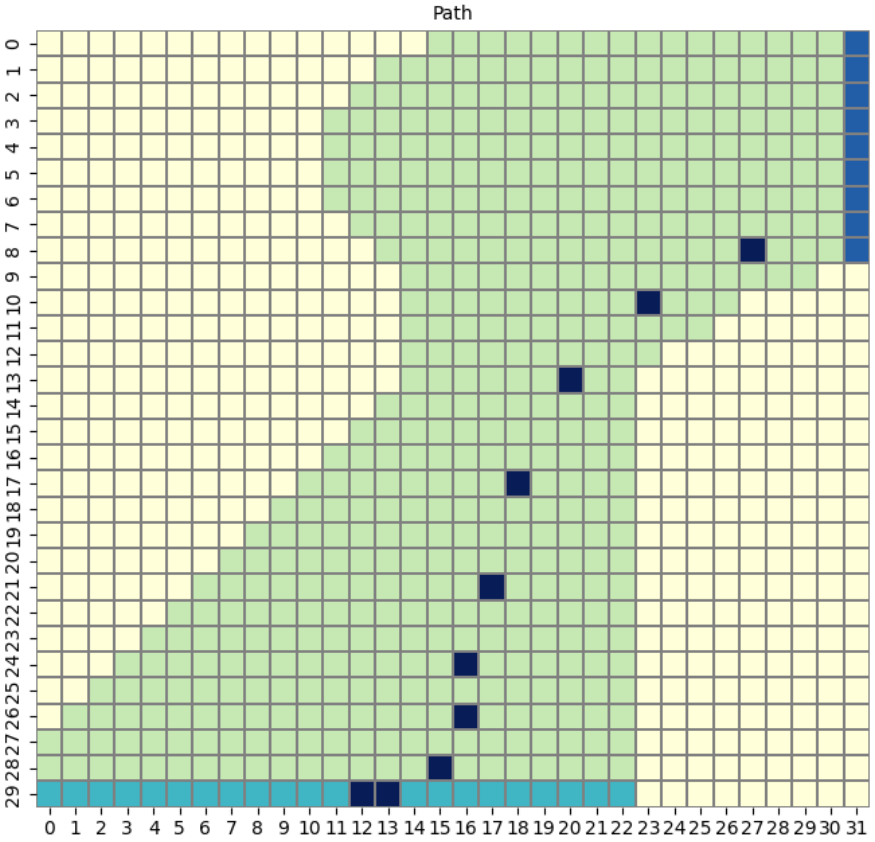
Average values over speeds.


Path
0
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
28
29
0
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
28
29
30
31

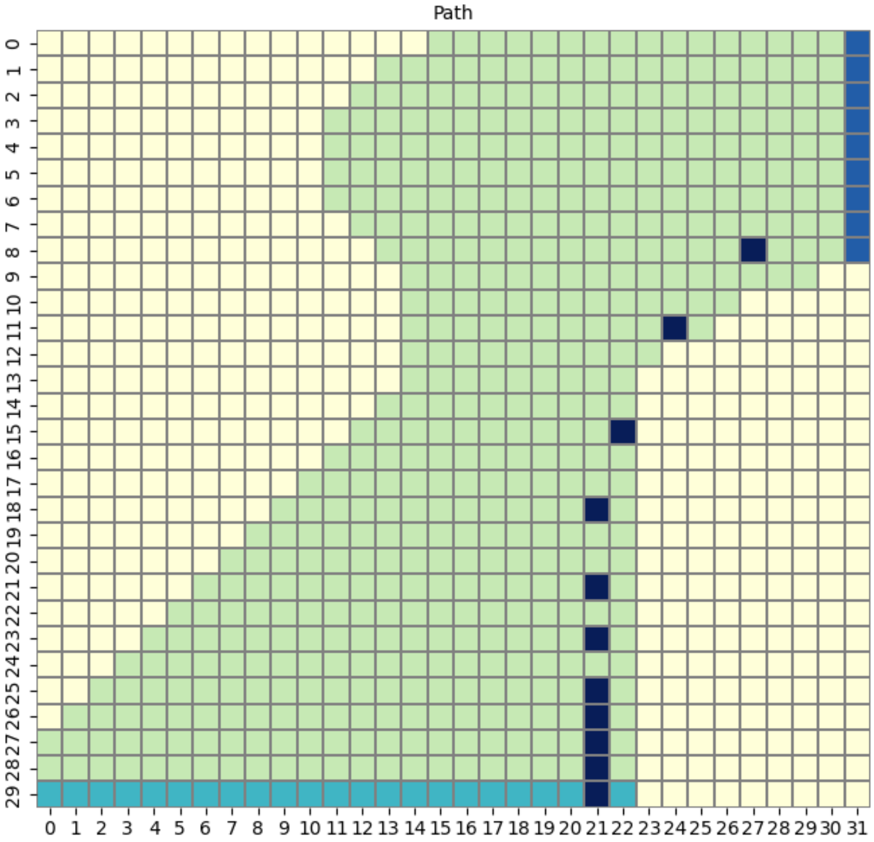
Demo path.


Path
0
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
28
29
0 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 28 29 30 31

Demo path.


Path
0
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
28
29
30
31

Demo path.



Demo path.

5.13 Question

Show the steps to derive (5.14) from (5.12).

Answer

$$\rho_{t:T-1}R_{t+1} = \frac{\pi(A_t|S_t)}{b(A_t|S_t)}\frac{\pi(A_{t+1}|S_{t+1})}{b(A_{t+1}|S_{t+1})}\cdots\frac{\pi(A_{T-1}|S_{T-1})}{b(A_{T-1}|S_{T-1})}R_{t+1} \tag{12}$$

$$E[\rho_{t:T-1}R_{t+1}] = E[\frac{\pi(A_t|S_t)}{b(A_t|S_t)}\frac{\pi(A_{t+1}|S_{t+1})}{b(A_{t+1}|S_{t+1})}\cdots\frac{\pi(A_{T-1}|S_{T-1})}{b(A_{T-1}|S_{T-1})}R_{t+1}] \tag{13}$$

R_{t+1} depends only to time step t thus we can write in $E[XY] = E[X] * E[Y]$ form.

$$E[\rho_{t:T-1}R_{t+1}] = E[\frac{\pi(A_t|S_t)}{b(A_t|S_t)}R_{t+1}]E[\frac{\pi(A_{t+1}|S_{t+1})}{b(A_{t+1}|S_{t+1})}\dots\frac{\pi(A_{T-1}|S_{T-1})}{b(A_{T-1}|S_{T-1})}] \quad (14)$$

More over, π or b does not depend on previous timestamp.

$$E[\rho_{t:T-1}R_{t+1}] = E[\frac{\pi(A_t|S_t)}{b(A_t|S_t)}R_{t+1}]E[\frac{\pi(A_{t+1}|S_{t+1})}{b(A_{t+1}|S_{t+1})}]\dots E[\frac{\pi(A_{T-1}|S_{T-1})}{b(A_{T-1}|S_{T-1})}] \quad (15)$$

Then we have $E[\frac{\pi(A_k|S_k)}{b(A_k|S_k)})] = 1$

$$E[\rho_{t:T-1}R_{t+1}] = E[\frac{\pi(A_t|S_t)}{b(A_t|S_t)}R_{t+1}] * 1\dots 1 \quad (16)$$

$$E[\rho_{t:T-1}R_{t+1}] = E[\rho_{t:t}R_{t+1}] \quad (17)$$

5.14 Question

Modify the algorithm for off-policy Monte Carlo control (page 111) to use the idea of the truncated weighted-average estimator (5.10). Note that you will first need to convert this equation to action values.

Answer

Weighted importance-sampling estimator:

$$V(s) = \frac{\sum_{t\in\tau(s)}[(1-\gamma)\sum_{h=t+1}^{T(t)-1}\gamma^{h-t-1}\rho_{t:h-1}\bar{G}_{t:h} + \gamma^{T(t)-t-1}\rho_{t:T(t)-1}\bar{G}_{t:T(t)}]}{\sum_{t\in\tau(s)}[(1-\gamma)\sum_{h=t+1}^{T(t)-1}\gamma^{h-t-1}\rho_{t:h-1} + \gamma^{T(t)-t-1}\rho_{t:T(t)-1}]} \quad (18)$$

where $\tau(s)$ is the set of all time steps in which state s is visited. We can turn this into action values:

$$Q(s,a) = \frac{\sum_{t\in\tau(s,a)}[(1-\gamma)\sum_{h=t+1}^{T(t)-1}\gamma^{h-t-1}\rho_{t:h-1}\bar{G}_{t:h} + \gamma^{T(t)-t-1}\rho_{t:T(t)-1}\bar{G}_{t:T(t)}]}{\sum_{t\in\tau(s,a)}[(1-\gamma)\sum_{h=t+1}^{T(t)-1}\gamma^{h-t-1}\rho_{t:h-1} + \gamma^{T(t)-t-1}\rho_{t:T(t)-1}]} \quad (19)$$

where $\tau(s, a)$ is the set of all time steps in which state s is visited and action a is selected.

We can turn this expression into update rule by following steps similar to Question 5.10 .

$$C_n = \sum_{k=1}^{n} B_k \tag{20}$$

$$Q_{n+1} = \frac{\sum_{k=1}^{n} A_k}{\sum_{k=1}^{n} B_k} = \frac{\sum_{k=1}^{n-1} A_k + A_n}{\sum_{k=1}^{n} B_k} = \frac{\frac{\sum_{k=1}^{n-1} B_k}{\sum_{k=1}^{n-1} B_k} \sum_{k=1}^{n-1} A_k + A_n}{\sum_{k=1}^{n} B_k} = \frac{Q_n \sum_{k=1}^{n-1} B_k + A_n}{\sum_{k=1}^{n} B_k} \tag{21}$$

$$Q_{n+1} = \frac{1}{C_n}(Q_n \sum_{k=1}^{n-1} B_k + A_n) = \frac{1}{C_n}(Q_n(\sum_{k=1}^{n} B_k - B_n) + A_n) = \frac{1}{C_n}(Q_n(C_n - B_n) + A_n) \tag{22}$$

$$Q_{n+1} = Q_n + \frac{A_n - Q_n B_n}{C_n} \tag{23}$$

Turn the update rule into incremental component updates.

C_n in recursive form

$$C_n = C_{n+1} + B_{n+1} \tag{24}$$

A_n in recursive form

$$A_n = \sum_{t \in \tau(s,a)} [(1-\gamma) \sum_{h=t+1}^{T(t)-1} \gamma^{h-t-1} \rho_{t:h-1} \bar{G}_{t:h} + \gamma^{T(t)-t-1} \rho_{t:T(t)-1} \bar{G}_{t:T(t)}] \tag{25}$$

Where:

$$a_t = \sum_{h=t+1}^{T(t)-1} \gamma^{h-t-1} \rho_{t:h-1} \bar{G}_{t:h} \tag{26}$$

$$w_t = \rho_{t:T(t)-1} \tag{27}$$

$$\bar{G}_t = \bar{G}_{t:T(t)} \tag{28}$$

$$A_n = \sum_{t \in \tau(s,a)} [(1-\gamma)a_t + \gamma^{T(t)-t-1} w_t \bar{G}_t] \tag{29}$$

B_n in recursive form

$$B_n = \sum_{t \in \tau(s,a)} [(1-\gamma) \sum_{h=t+1}^{T(t)-1} \gamma^{h-t-1} \rho_{t:h-1} + \gamma^{T(t)-t-1} \rho_{t:T(t)-1}] \tag{30}$$

Where:

$$b_t = \sum_{h=t+1}^{T(t)-1} \gamma^{h-t-1} \rho_{t:h-1} \tag{31}$$

$$B_n = \sum_{t \in \tau(s,a)} [(1-\gamma)b_t + \gamma^{T(t)-t-1} w_t] \tag{32}$$

b_t can be expanded

$$b_t = \gamma^0 \rho_{t:t} + \gamma^1 \rho_{t:t+1} + \gamma^2 \rho_{t:t+2} + \cdots + \gamma^{T-t-2} \rho_{t:T-2} \tag{33}$$

$$b_t = \gamma^0 \frac{\pi(s_t|a_t)}{b(s_t|a_t)} + \gamma^1 \frac{\pi(s_t|a_t)}{b(s_t|a_t)} \frac{\pi(s_{t+1}|a_{t+1})}{b(s_{t+1}|a_{t+1})} + \gamma^2 \frac{\pi(s_t|a_t)}{b(s_t|a_t)} \frac{\pi(s_{t+1}|a_{t+1})}{b(s_{t+1}|a_{t+1})} \frac{\pi(s_{t+2}|a_{t+2})}{b(s_{t+2}|a_{t+2})} + \dots \tag{34}$$

b_t can be written recursively:

$$b_t = b_{t+1} * \gamma \frac{\pi(s_t|a_t)}{b(s_t|a_t)} + \frac{\pi(s_t|a_t)}{b(s_t|a_t)} \text{ where } b_T = 0 \tag{35}$$

$\bar{G}_t$ can be written recursively

$$\bar{G}_t = \bar{G}_{t:T} = R_{t+1} + \bar{G}_{t+1:T} \tag{36}$$

w_t can be written recursively

$$w_t = \rho_{t:t} + w_{t+1} \tag{37}$$

a_t can be expanded

$$a_t = \gamma^0 \rho_{t:t} \bar{G}_{t:t+1} + \gamma^1 \rho_{t:t+1} \bar{G}_{t:t+2} + \gamma^2 \rho_{t:t+2} \bar{G}_{t:t+3} + \dots + \gamma^{T-t-2} \rho_{t:T-2} \bar{G}_{t:T-1} \tag{38}$$

$$\begin{aligned} a_t = \gamma^0 \frac{\pi(s_t|a_t)}{b(s_t|a_t)} R_{t+1} + \gamma^1 \frac{\pi(s_t|a_t)}{b(s_t|a_t)} \frac{\pi(s_{t+1}|a_{t+1})}{b(s_{t+1}|a_{t+1})} (R_{t+1} + R_{t+2}) + \\ \gamma^2 \frac{\pi(s_t|a_t)}{b(s_t|a_t)} \frac{\pi(s_{t+1}|a_{t+1})}{b(s_{t+1}|a_{t+1})} \frac{\pi(s_{t+2}|a_{t+2})}{b(s_{t+2}|a_{t+2})} (R_{t+1} + R_{t+2} + R_{t+3}) + \dots \end{aligned}$$

a_t can be written recursively:

$$a_T = a_{T-1} = 0 \tag{39}$$

$$a_{T-2} = \frac{\pi(s_{T-2}|a_{T-2})}{b(s_{T-2}|a_{T-2})} R_{T-1} = \frac{\pi(s_{T-2}|a_{T-2})}{b(s_{T-2}|a_{T-2})} (R_{T-1} * [1] + \gamma a_{T-1}) \tag{40}$$

$$a_{T-3} = \frac{\pi(s_{T-3}|a_{T-3})}{b(s_{T-3}|a_{T-3})} (R_{T-2} * [1 + \gamma \frac{\pi(s_{T-2}|a_{T-2})}{b(s_{T-2}|a_{T-2})}] + \gamma a_{T-2}) \tag{41}$$

$$a_{T-4} = \frac{\pi(s_{T-4}|a_{T-4})}{b(s_{T-4}|a_{T-4})} (R_{T-3} * [1 + \gamma \frac{\pi(s_{T-3}|a_{T-3})}{b(s_{T-3}|a_{T-3})} + \gamma^2 \frac{\pi(s_{T-2}|a_{T-2})}{b(s_{T-2}|a_{T-2})}] + \gamma a_{T-3}) \tag{42}$$

The expression within the square brackets can be simplified to:

$$M_t = 1 + \sum_{k=2}^{T-t-1} \gamma^{T-t-k} \frac{\pi(s_{T-k}|a_{T-k})}{b(s_{T-k}|a_{T-k})} \tag{43}$$

Right side of plus one can be further simplified to:

$$m_t = \sum_{k=2}^{T-t-1} \gamma^{T-t-k} \frac{\pi(s_{T-k}|a_{T-k})}{b(s_{T-k}|a_{T-k})} \tag{44}$$

m_t can be written recursively:

$$m_t = \gamma(\frac{\pi(s_t|a_t)}{b(s_t|a_t)} + m_{t+1}) \text{ where } m_{T-2} = 0 \tag{45}$$

Final recursive expression for a_t would be:

$$a_t = \frac{\pi(s_t|a_t)}{b(s_t|a_t)}(R_{t+1}(1 + m_{t+1}) + \gamma a_{t+1}) \tag{46}$$

The final algorithm would look like:

```
Initialize

$Q(s, a) = 0$ for all s,a.
$C(s, a) = 0$ for all s,a.
$\pi(s) = a_0$

Loop forever

b=any soft policy
Generate and episode using $b : S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$
$\bar{G} = R_t$
$w = 0$
$a = 0$
$b = 0$
$m = 0$

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

$\bar{G} = R_{t+1} + \bar{G}$
$A = (1 - \gamma)a + \gamma^{T-t-1} w\bar{G}$
$B = (1 - \gamma)b + \gamma^{T(t)-t-1} w$
$C = C + B$
$Q = Q + \frac{A-QB}{C}$
$\pi(S_t) = \operatorname{argmax}_a Q(S_t, a)$
if $A_t \neq \pi(S_t)$ then exit inner loop
$m = \gamma(\frac{1}{b(s_t|a_t)} + m)$
$a = \frac{1}{b(s_t|a_t)}(R_{t+1}(1 + m) + \gamma a)$
$b = b * \gamma\frac{1}{b(s_t|a_t)} + \frac{1}{b(s_t|a_t)}$
$w = \frac{1}{b(s_t|a_t)} + w$
```