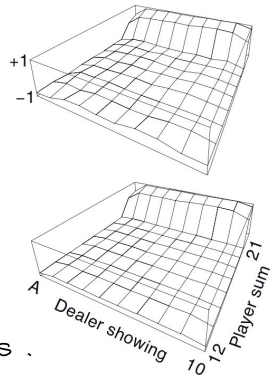
Exercise 5.1 Consider the diagrams on the right in Figure 5.1. Why does the estimated value function jump up for the last two rows in the rear? Why does it drop off for the whole last row on the left? Why are the frontmost values higher in the upper diagrams than in the lower? ☑

Player sums nearby 20 or 21 means he's very likely to win because the dealer is unlikely to get a higher value without going bust. With 21 the worst outcome is a draw when the dealer gets a Blackjack.



Drop-off on the left is caused by the dealer showing an ace. Conditioned on this the dealer is very likely ($4/13$) to get a Blackjack or a very high score. Ace also lowers the risk of going bust by additional hits.

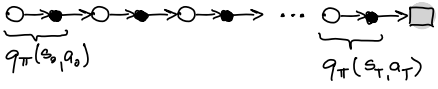
Upper diagram shows value function with usable ace. Having usable ace reduces the risk of going bust by hitting and therefore these states have higher expected probability of winning.

Exercise 5.2 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? ☑

Results would be the same. Reason is that during an episode of Blackjack every state (i.e. hand value) is visited only once. Single visit is guaranteed thanks to the ace usability flag being part of the state. Consider the following example episode that would visit a state twice without it:

$\underbrace{\{A,9\}}_{\text{cards}} : \underbrace{20, \text{usable}}_{\text{state}} \xrightarrow{\text{hit: 10}} \underbrace{\{A,9,10\}}_{\text{cards}} : \underbrace{20, \text{not usable}}_{\text{state}}$

Exercise 5.3 What is the backup diagram for Monte Carlo estimation of q_π? ☑



Backup diagram shows the node to be updated at the top and all the values needed for the update as leafs.

Exercise 5.4 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.

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 $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$
  $N(s,a) \leftarrow 0$ 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~~ $Returns(S_t, A_t) \leftarrow Returns(S_t, A_t) + (G - Returns(S_t, A_t)) / N(S_t, A_t)$
      $Q(S_t, A_t) \leftarrow$ ~~average~~$(Returns(S_t, A_t))$
      $\pi(S_t) \leftarrow \arg\max_a Q(S_t, a)$

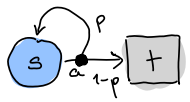
$N(S_t, A_t) \leftarrow N(S_t, A_t) + 1$

Alternatively, without using Returns:
$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + (G - Q(S_t, A_t)) / N(S_t, A_t)$

Exercise 5.5 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?

First visit updates the Returns list for the non-terminal state only once, at the first time-step:

$Q_{FV}(s,a) = average([10]) = 10$



Every visit updates the result every time-step with an increasing return value:

$Q_{EV}(s,a) = average([1, 2, 3, \ldots, 10]) = 5\,1/2$

Exercise 5.6 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?

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

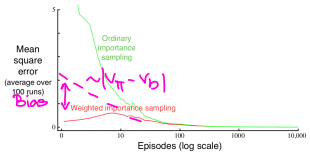
$\mathcal{T}(s,a)$ ... all timesteps when visit to state s was followed by action a

$T(t)$ ... timestep of termination of the current episode

$\rho_{t:T-1}$ ... importance sampling ratio $\prod_{k=t}^{T-1} \frac{\pi(A_k|S_k)}{b(A_k|S_k)}$

Exercise 5.7 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? ☑

WIS has initial bias and outputs values closer to $V_b(s)$ and over time and additional episodes it shifts closer to $V_\pi(s)$. Because the two policies are very different these two values are not close and output close to $V_b(s)$ doesn't match the acutual performance of π.



Exercise 5.8 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? ☑

Yes. The variance of the estimator would still be infinite. Both first-visit and every-visit MC method in this particular case return the same value of G_0. Therefore the argumentation on page 108 is valid in both situations as it's independent of G_0. Because there's a non-zero reward only at the very end of each episode the estimates are equal:

G_0^{FV} = average([1]) $\quad G_0^{EV}$ = average([1, 1, 1, ... , 1])

Overall count depends on the episode duration but average value is always 1.

Exercise 5.9 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. ☑

First-visit MC prediction, for estimating $V \approx v_\pi$

Input: a policy π to be evaluated
Initialize:
  $V(s) \in \mathbb{R}$, ~~arbitrarily~~ 0, for all $s \in \mathcal{S}$
  ~~$Returns(s) \leftarrow$ an empty list, for all $s \in \mathcal{S}$~~
  $N(s) \leftarrow 0$ 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 $Returns(S_t)$~~ $N(S_t) \leftarrow N(S_t) + 1$
      $V(S_t) \leftarrow$ ~~average($Returns(S_t)$)~~ $V(S_t) + (G - V(S_t))/N(S_t)$

Exercise 5.10 Derive the weighted-average update rule (5.8) from (5.7). Follow the pattern of the derivation of the unweighted rule (2.3). ☑

$$V_{n+1} = \frac{\sum_{k=0}^{n} W_k G_k}{\sum_{k=0}^{n} W_k} = \frac{1}{\sum_{k=0}^{n} W_k}\left(W_n G_n + \underbrace{\sum_{k=0}^{n-1} W_k G_k}_{=\left(\sum_{k=0}^{n-1} W_k\right) V_n}\right) = V_n + \frac{W_n}{\underbrace{\sum_{k=0}^{n} W_k}_{=C_n}}\left(G_n - V_n\right) =$$

C_n can be updated in the same way:

$$C_n = \sum_{k=0}^{n} W_k = W_n + \underbrace{\sum_{k=0}^{n-1} W_k}_{C_{n-1}} = \qquad = \sum_{k=0}^{n} W_k - W_n$$
$$= W_n + C_{n-1}$$

Exercise 5.11 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? ☑

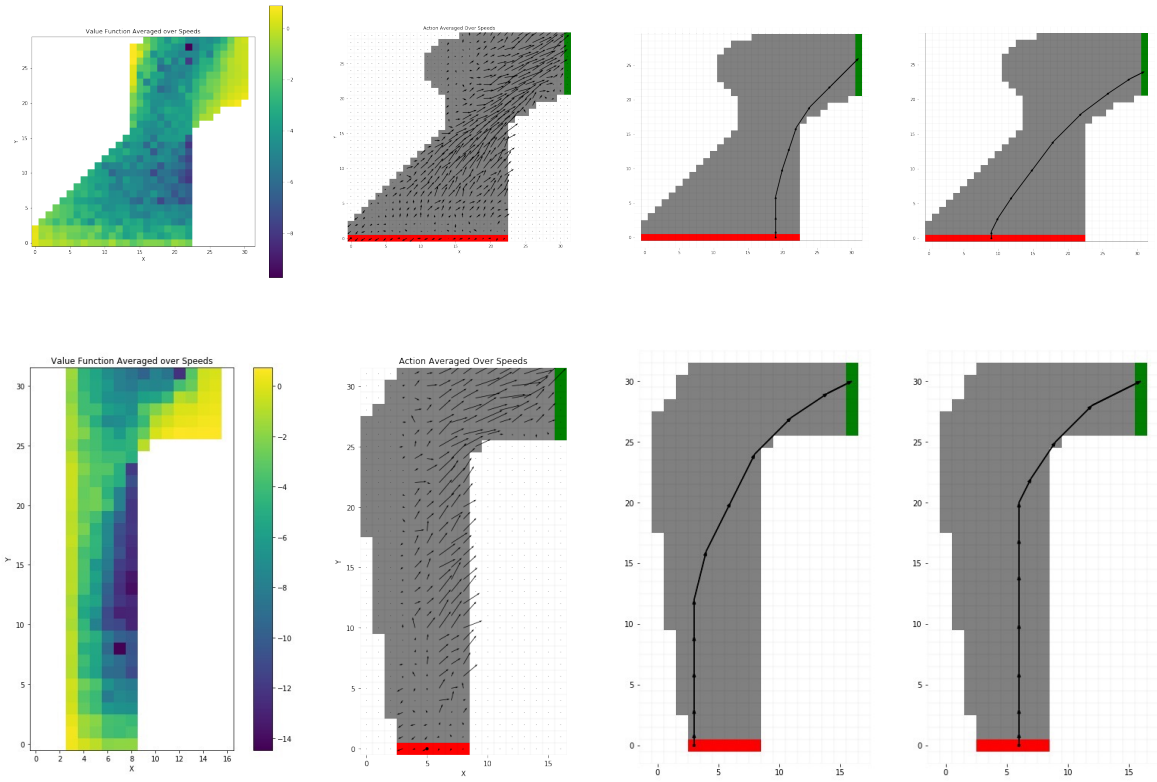
Off-policy MC control, for estimating $\pi \approx \pi_*$

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, $t = T-1, T-2, \ldots, 0$:
    $G \leftarrow \gamma G + R_{t+1}$
    $C(S_t, A_t) \leftarrow C(S_t, A_t) + W$ (W=0 after $A_t \neq \pi(S_t)$)
    $Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \frac{W}{C(S_t, A_t)}\left[G - Q(S_t, A_t)\right]$
    $\pi(S_t) \leftarrow \arg\max_a Q(S_t, a)$ (with ties broken consistently)
    If $A_t \neq \pi(S_t)$ then exit inner Loop (proceed to next episode) ← skip to the next episode
    $W \leftarrow W \frac{1}{b(A_t|S_t)}$ ← 0 for all t after $A_t \neq \pi(S_t)$

Because π is a deterministic policy (i.e. zero everywhere except a single action with probability one) exiting the loop earlier before the importance sampling ratio update achieves the same result. The equivalent operation would be setting $W \leftarrow 0$ which means all further update to Q or C are identities.
This also means that only the very last segment of the episode where taken actions agree with π is used for learning and the rest of the episode data is uselessly thrown away. After the divergence point the importance ratio product contains 0 because the scenario is not possible when following policy π.

Exercise 5.12: 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. In our simplified racetrack, the car is at one of a discrete set of grid positions, the cells in the diagram. The velocity is also discrete, a number of grid cells moved horizontally and vertically per time step. The actions are increments to the velocity components. Each may be changed by $+1$, -1, or 0 in each step, for a total of nine (3×3) actions. Both velocity components are restricted to be nonnegative and less than 5, and they cannot both be zero except at the starting line. Each episode begins in one of the randomly selected start states with both velocity components zero and ends when the car crosses the finish line. The rewards are -1 for each step until the car crosses the finish line. If the car hits the track boundary, it is moved back to a random position on the starting line, both velocity components are reduced to zero, and the episode continues. Before updating the car's location at each time step, check to see if the projected path of the car intersects the track boundary. If it intersects the finish line, the episode ends; if it intersects anywhere else, the car is considered to have hit the track boundary and is sent back to the starting line. To make the task more challenging, with probability 0.1 at each time step the velocity increments are both zero, independently of the intended increments. 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 off for these trajectories).



Exercise 5.13 Show the steps to derive (5.14) from (5.12).

$$\mathbb{E}_b\left[\rho_{t:T-1} R_{t+1}\right] = \sum_{\forall r \forall s' \forall a_t \cdots \forall a_{T-1}} r\, p(s', r \mid a_t, s_t)\, b(a_t \mid s_t) \cdots b(a_{T-1} \mid s_{T-1}) \frac{\pi(a_t \mid s_t) \cdots \pi(a_{T-1} \mid s_{T-1})}{b(a_t \mid s_t) \cdots b(a_{T-1} \mid s_{T-1})} =$$

$$= \sum_{\forall r \forall s' \forall a_t} r\, p(s', r \mid s_t, a_t)\, b(a_t \mid s_t) \underbrace{\frac{\pi(a_t \mid s_t)}{b(a_t \mid s_t)}}_{\rho_{t:t}} \underbrace{\sum_{\forall a_{t+1} \cdots \forall a_{T-1}} \pi(a_{t+1} \mid s_{t+1}) \cdots \pi(a_{T-1} \mid s_{T-1})}_{\sum_{\forall a_{t+1}} \pi(a_{t+1} \mid s_{t+1}) \cdots \sum_{\forall a_{T-1}} \pi(a_{T-1} \mid s_{T-1})}$$

$$\underbrace{\qquad\qquad}_{\mathbb{E}_b\left[R_{t+1} \rho_{t:t}\right]} \qquad \underbrace{\qquad}_{=1} \cdots \underbrace{\qquad}_{=1}$$

Exercise 5.14 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.

Converting the truncated weighted importance formula is straightforward:

$$Q(s,a) = \frac{\sum_{t \in \mathcal{T}(s,a)} \left((1-\gamma) \sum_{h=t+1}^{T(t)-1} \gamma^{h-t-1} \rho_{t:h-1} \bar{G}_{t:h} \; + \; \gamma^{T(t)-t-1} \rho_{t:T(t)-1} \bar{G}_{t:T(t)} \right)}{\sum_{t \in \mathcal{T}(s,a)} \left((1-\gamma) \sum_{h=t+1}^{T(t)-1} \gamma^{h-t-1} \rho_{t:h-1} \; + \; \gamma^{T(t)-t-1} \rho_{t:T(t)-1} \right)}$$

$\mathcal{T}(s,a)$... all timesteps when visit to s was followed by action a. (or only the 1st visit)

Truncated weighted importance sampling estimator formula has a general shape that allows for incremental updates as new episodes arrive during training:

$$Q_{n+1} = \frac{\sum_{k=1}^{n} A_k}{\sum_{k=1}^{n} B_k} = \frac{\overbrace{\sum_{k=1}^{n-1} A_k}^{\sim Q_{n-1}} + A}{\sum_{k=1}^{n} B_k} = \frac{\left(\sum_{k=1}^{n-1} B_k\right) Q_{n-1} + A}{\sum_{k=1}^{n} B_k} = Q_n + \frac{A_n - B_n Q_n}{\sum_{k=1}^{n} B_k}$$

(target: $A_n - B_n Q_n$; weight: $\sum_{k=1}^{n} B_k$)

So the problem now is how to efficiently calculate A_k and B_k. The sums to evaluate them depend on the previous history of the currently evaluated episode. Therefore it's worth investigating whether their structure allows for incremental implementation similar to Q update rule above.

$$A_n = \sum_{t \in \mathcal{T}(s,a)} \left((1-\gamma) \overbrace{\sum_{h=t+1}^{T(t)-1} \gamma^{h-t-1} \rho_{t:h-1} \bar{G}_{t:h}}^{\alpha_t} \; + \; \overbrace{\gamma^{T(t)-t-1} \rho_{t:T(t)-1}}^{W_t} \overbrace{\bar{G}_{t:T(t)}}^{\bar{G}_t} \right)$$

$$B_n = \sum_{t \in \mathcal{T}(s,a)} \left((1-\gamma) \overbrace{\sum_{h=t+1}^{T(t)-1} \gamma^{h-t-1} \rho_{t:h-1}}^{\beta_t} \; + \; \overbrace{\gamma^{T(t)-t-1} \rho_{t:T(t)-1}}^{W_t} \right)$$

W_t and $\bar{G}_t$ can be tracked in exactly the same way as in the original algorithm.

$$\bar{G}_{t-1} = \bar{G}_t + R_{t+1} \quad [\bar{G}_T = R_T] \qquad W_{t-1} = W_t \gamma \overbrace{\frac{\pi(A_t|S_t)}{b(A_t|S_t)}}^{=1^*} \quad [W_T = 1]$$

Incremenatal tracking of a_t and b_t is trickier but also possible with the following identities $\rho_{x:y}\rho_{x+1:z} = \rho_{x:z}$, $\bar{G}_{x:y} + \bar{G}_{y+1:z} = \bar{G}_{x:z}$:

$$\beta_{t-1} = \frac{\pi(A_t|S_t)^*}{b(A_t|S_t)}\gamma\beta_t + W_t \quad [\beta_T = 1]$$

$$\alpha_{t-1} = \frac{\pi(A_t|S_t)^*}{b(A_t|S_t)}\gamma\alpha_t + \beta_t R_{t+1} + \frac{\pi(A_t|S_t)^*}{b(A_t|S_t)}\gamma R_{t+1} + W_t\bar{G}_t \quad [\alpha_T = R_T]$$

Incremental updates to A_n, B_n and C_n are:

$$A_n = (1-\gamma)\alpha_t + \gamma^{T-t-1} W_t \bar{G}_t$$

$$B_n = (1-\gamma)\beta_t + \gamma^{T-t-1} W_t$$

$$C_{n-1} = C_n + B_n \quad [C_0 = 0]$$

t ... index that goes backward throughout the episode timesteps (reset every episode)

n ... index of updates to Q (persistent across episodes)

Finally, update to Q_n is:

$$Q_{n-1} = Q_n + \frac{1}{C_n}(A_n - B_n Q_n) \quad [Q_0 = 0]$$

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