Name: Alfred Cueva
Student No.: 2020-12788

Exercise 3

Data Science & Reinforcement Learning

Spring 2023

1. 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 $Returns(S_t, A_t)$
$Q(S_t, A_t) \leftarrow$ average$(Returns(S_t, A_t))$
$\pi(S_t) \leftarrow \arg\max_a Q(S_t, a)$

since $Q_{n+1} = Q_n + \frac{1}{n}(R_n - Q_n) = \frac{1}{n}\sum_{i=1}^{n} R_t$

$n = t$ ← immediate

$Q_n(S_t, A_t) = \frac{1}{n}\sum_{i=1}^{n} G_t(S_t, A_t) = \frac{1}{n}\left(G_n(S_t, A_t) + \sum_{i=1}^{n-1} G_t(S_t, A_t)\right)$

$Q_n(S_t, A_t) = Q_{n-1}(S_t, A_t) + \frac{1}{n}\left(G_n(S_t, A_t) - Q_{n-1}(S_t, A_t)\right)$

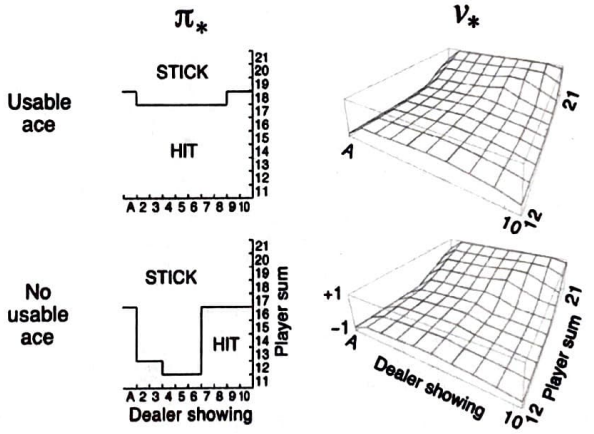
$N(S,a) \leftarrow 0$ for all $S \in \mathcal{S}$, $a \in A(s)$
$G \leftarrow 0$
Loop for each step of episode, $t = T-1, T-2, \ldots, 0$:
$G \leftarrow \gamma G + R_{t+1}$
unless the pair S_t, A_t appears in $S_0, A_0 \ldots S_{T-1}, A_{T-1}$:
$N(S_t, A_t) \leftarrow N(S_t, A_t) + 1$

$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \frac{G - Q(S_t, A_t)}{N(S_t, A_t)}$

$\pi(S_t) \leftarrow \arg\max_a Q(S_t, a)$

2. **(Blackjack)**[1] Playing blackjack is naturally formulated as an episodic finite MDP. Each game of blackjack is an episode. Rewards of +1, −1, and 0 are given for winning, losing, and drawing, respectively. All rewards within a game are zero, and we do not discount ($\gamma = 1$); therefore these terminal rewards are also the returns. The player's actions are to hit or to stick. The states depend on the player's cards and the dealer's showing card. We assume that cards are dealt from an infinite deck (i.e., with replacement) so that there is no advantage to keeping track of the cards already dealt. If the player holds an ace that he could count as 11 without going bust, then the ace is said to be usable. In this case it is always counted as 11 because counting it as 1 would make the sum 11 or less, in which case there is no decision to be made because, obviously, the player should always hit. Thus, the player makes decisions on the basis of three variables: his current sum (12–21), the dealer's one showing card (ace–10), and whether or not he holds a usable ace[2]. This makes for a total of 200 states.

   It is straightforward to apply Monte Carlo ES to blackjack. Because the episodes are all simulated games, it is easy to arrange for exploring starts that include all possibilities. In this case one simply picks the dealer's cards, the player's sum, and whether or not the player has a usable ace, all at random with equal probability. As the initial policy we use the policy which sticks only on 20 or 21. The initial action-value function can be zero for all state–action pairs. The figure below shows the optimal policy for blackjack found by Monte Carlo ES.



   What is the optimal action when the player' sum is 15 and the dealer's showing 5 with *usable* ace? How about without *usable* ace? What is the usability of ace used for in MDP? Hint, is it state, action, transition, or policy?

   - optimal action is hit for usable ace and stick without the usable ace while the usability of the ace is state.

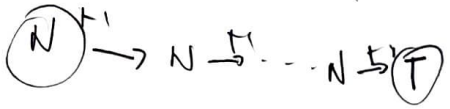
[1] The object of the popular casino card game of blackjack is to obtain cards the sum of whose numerical values is as great as possible without exceeding 21. All face cards count as 10, and an ace can count either as 1 or as 11. We consider the version in which each player competes independently against the dealer. The game begins with two cards dealt to both dealer and player. One of the dealer's cards is face up and the other is face down. If the player has 21 immediately (an ace and a 10-card), it is called a natural. He then wins unless the dealer also has a natural, in which case the game is a draw. If the player does not have a natural, then he can request additional cards, one by one (hits), until he either stops (sticks) or exceeds 21 (goes bust). If he goes bust, he loses; if he sticks, then it becomes the dealer's turn. The dealer hits or sticks according to a fixed strategy without choice: he sticks on any sum of 17 or greater, and hits otherwise. If the dealer goes bust, then the player wins; otherwise, the outcome—win, lose, or draw—is determined by whose final sum is closer to 21.

[2] Aces can count as 11 or 1, and if counted as 11, the ace is called usable

3. 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?

$|T(s)| = \sum_{t \in T(s)} P_t : T(t)-t = 10$

$P_t : T(t)-t = 1$



first-visit: update return list for non terminal state only once. for first time step.

V_s = average(lists) = 10

every visits.

$V_s = \frac{1}{10}(1 + \dots + 10) = \frac{10 \cdot 11}{2 \cdot 10} = 5.5$

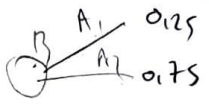
4. Let $\rho_t = \frac{\pi(A_t|S_t)}{b(A_t|S_t)}$.

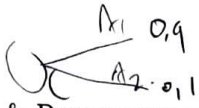
(a) Verify that $\mathbb{E}_b[\rho_t | S_t = s] = 1$.

$E_b[\rho_t | S_t = s] = \sum_{A_t \in A(s)} \pi(A_t|S_t) \cdot b(A_t|S_t) = 1$ running all probs

(b) Verify that $\mathbb{E}_b[\rho_t R_{t+1} | S_t = s] = \mathbb{E}_\pi[R_{t+1} | S_t = s]$.

$LHS = \sum_{A_t \in A(s)} \frac{\pi(A_t|S_t)}{b(A_t|S_t)} \cdot b(A_t|S_t) R_t = \sum_{A_t \in A(s)} \pi(A_t|S_t) R_t = E_\pi[R_{t+1} | S_t = s]$ ∎

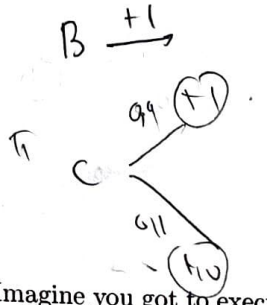




5. Off-policy Monte Carlo prediction allows us to use sample trajectories to estimate the value function for a policy that may be different than the one used to generate the data. Consider the following MDP, with two states B and C, with 1 action in state B and two actions in state C, with $\gamma = 1.0$. Assume the target policy π has $\pi(A = 1|C) = 0.9$ and $\pi(A = 2|C) = 0.1$, and that the behaviour policy b has $b(A = 1|C) = 0.25$ and $b(A = 2|C) = 0.75$.

deterministic MDP

(a) What are the true values $V_\pi(B)$, and $V_\pi(C)$?



$V_\pi(B) = 0,9 \times 1 + 0,1 \times 10 = 1,9$

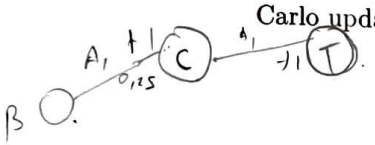
$V_H(C) = 1 + V_H(B)$

$= 2,9$

$G_B = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} = 1 + 1 = 2$

$G_C = 1$

(b) Imagine you got to execute π in the environment for one episode, and observed the episode trajectory $S_0 = B, A_0 = 1, R_1 = 1, S_1 = C, A_1 = 1, R_2 = 1$. What is the return for B for this episode? Additionally, what are the value estimates $V_\pi(B)$ and $V_\pi(C)$, using this one episode with Monte Carlo updates?

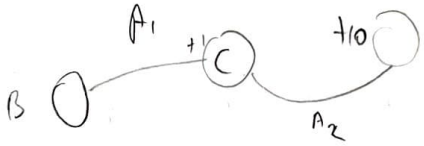


π is behaviour then sum for every:

$V_\pi(B) = 2$

$V_H(C) = 1$

(c) Suppose you followed the behaviour policy b instead and observed the episode trajectory $S_0 = B, A_0 = 1, R_1 = 1, S_1 = C, A_1 = 2, R_2 = 10$. What is the return for B for this episode? Additionally, what are the value estimates $V_\pi(B)$ and $V_\pi(C)$, using this one episode with Monte Carlo updates?



$G_B = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} = 1 + 1 \cdot (10) = 11$

$G_C = 10$

$V_H(C) = \frac{\frac{0,1}{0,75} \times 10}{1} = 1,\overline{33}$ OIS

$V_\pi(B) = \frac{1 \times \frac{0,1}{0,75} \times 11}{1} = 1,47$ OIS