Reinforcement Learning :-

A-3 - Theory

Donal Loitam (AI21BTECH11009)

Ans 1(a)

Given the samples, we observe that [Using first-visit MC]

$N(A) = 5$ $\quad N(D) = 4$ (Not in 3rd sample)

$N(B) = 5$ $\quad N(E) = 5 = N(F)$

$N(C) = 5$

Let, $G_j(A)$:= denote discounted sum of rewards starting from the first-visit of (state A) in sample (trajectory) j.

$S(A)$:= denote the running sum of total returns

$S(A) = G_1(A) + G_2(A) + G_3(A) + G_4(A) + G_5(A)$

$= 14 + 15 + 17 + 16 + 15 = 77$

Sly, $S(B) = \sum_{i=1}^{5} G_i(B)$

$= 13 + 14 + 16 + 15 + 14 = 72$

$S(C) = 12 + 13 + 15 + 14 + 13 = 67$

$S(D) = 12 + 12 + 0 + 12 + 11 = 47$

$S(E) = 11 + 11 + 11 + 10 + 10 = 52$

$S(F) = 10 + 10 + 10 + 10 + 9 = 49$

$\therefore V(A) = \frac{77}{5}$, $V(B) = \frac{72}{5}$, $V(C) = \frac{67}{5}$ Ans

$V(D) = \frac{47}{4}$, $V(E) = \frac{52}{5}$, $V(F) = \frac{49}{5}$

1)(a) $\boxed{V(G) = \frac{0}{5} = 0}$ $(\because S(G) = 0)$

Ans

1 (b) Since, states C, E, F are visited more than once in one (or) more given episodes / samples, they are likely to have different value estimates. But, we may still get the same value.

1 (c)

$$C \xrightarrow[\text{jump}]{4} E \xrightarrow[\text{right}]{1} F \xrightarrow[\text{left}]{-2} E \xrightarrow[\text{right}]{1} F$$

Given, $\boxed{\alpha = 0.7, \gamma = 1}$

We'll be using the Q-learning update rule,

(i) $Q(C, \text{jump}) = Q(C, \text{jump}) + 0.7\left(4 + 1 \max_a Q(E, a) - Q(C, \text{jump})\right)$

$= -10 + .7\left(4 + 1(-10) - (-10)\right)$

$= \boxed{-7.2}$

In general, the formula :-

$$\boxed{Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha_t \left[r_{t+1} + \gamma . \max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t)\right]}$$

(ii) $Q(E, \text{right}) = -10 + 0.7[1 + 1 \times (-10) - (-10)]$

$= \cancel{10}\ \boxed{-9.3}$

(iii) $Q(E, \text{left}) = -10 + 0.7[-2 + 1 \max(-9.3, -10) - (-10)]$

$= -10 + -0.91 = \boxed{-10.91}$

Cont'd

1(c) (iv) $Q(E, \text{right}) = -9.3 + 0.7\left[1 + 1 * \max(-10.91, -10) - (-9.3)\right]$

$= -9.3 + 0.21 = \boxed{-9.09}$

$Q(c,\text{left})$	$Q(c,\text{jump})$	$Q(E,\text{left})$	$Q(E,\text{right})$	$Q(F,\text{left})$	$Q(E,\text{right})$
-10	-10	-10	-10	-10	-10
-10	-7.2	-10	-10	-10	-10
-10	-7.2	-10	-9.3	-10	-10
-10	-7.2	-10	-9.3	-10.91	-10
-10	-7.2	-10	-9.09	-10.91	-10

1.(d) $\pi(c) = \underset{a}{\text{argmax}}\ Q(c,a) = \text{jump}$ $\left\{ \because \begin{array}{l} \text{jump} \to -7.2 \\ \text{left} \to -10 \end{array} \right.$

1(e) Robins Monroe Cond^n are :-

- $\boxed{\sum \alpha_t = \infty}$ and $\boxed{\sum \alpha_t^2 < \infty}$ (finite)

(i) $\alpha_t = \frac{1}{t}$

From calculus, we know that $\sum_{t=1}^{\infty} 1/t^p$ converges if $p > 1$.

$\sum \alpha_t = \sum 1/t = \infty$ [Harmonic series] [$\because p = 1$]

$\sum \alpha_t^2 = \sum \frac{1}{t^2} = c < \infty$ [$\because p = 2 > 1$]

$\boxed{\text{Note}}$: p-series test :-

The series $\sum_{n=1}^{\infty} \left(\frac{1}{n}\right)^p$ converges iff $\boxed{p > 1}$

P.T.O →

$$\boxed{\sum = \sum_{t=1}^{\infty}}$$

1(e) (ii) $\alpha_t = \frac{1}{t^2}$ $\qquad C = \text{const}$

$$\sum \alpha_t = \sum \frac{1}{t^2} = C < \infty \quad [\because p = 2 > 1]$$

So, $\alpha_t = \frac{1}{t}$ follows the condn

while $\alpha_t = \frac{1}{t^2}$ violates it.

1(f) Q-learning follows Generalised Policy Iteration to find the optimal policy. In the specific case of Q-learning, we use TD approximation method (~~the off-policy one~~) to evaluate optimal state-action function.

TD Appx:

$$\boxed{Q(s_t, a_t) := Q(s_t, a_t) + \alpha_t * [R(s_t, a_t, s_{t+1}) + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)]}$$

Where, the trajectory segment $(s_t, a_t, r_{t+1}, s_{t+1})$ is generated by the ε-greedy policy.

while, $\max_a Q(s_{t+1}, a) := a_{t+1}$ is greedy w.r.t Q. [being greedy at s_{t+1}]

So, our target is greedy w.r.t $Q(s, a)$

In the f(i), all the convergence criteria for Policy EVALUATION are satisfied [Given]

1(5) Also, the only difference in the question is that we are sampling a_t [from s_t] by following a fixed policy π with prob = 0.5 (or) Choose an action uniform randomly.

But, because our target policy is greedy, $Q(s_t, a_t)$ will improve for all (s_t, a_t) as there is randomness in choosing action and all s-a pairs are visited only.. So, it will eventually converge to the optimal Q value f^n.

(ii) SARSA is an on-policy, implies it uses same policy to interact with environment and learn from it too. In the given scenario, the SARSA agent will not reach the optimal Q-value f^n, since there would be no improvement in the Q-values [since the target is not improving]

Because we were following ϵ-greedy over Q to choose action 'a_{t+1}' in SARSA, but here we are following a fixed policy or random [no-greedy] & we already proved that For an policy π, the ϵ-greedy policy π' w.r.t Q^{π} is an improvement over π. [Here, no sign of greedy]