ASSIGNMENT - 2

Que - 1)

Arms : $\{1, 2, \ldots 10\}$ a_e : even arm a_o : odd Arm

$$P[a_e] = 2\,P[a_o]$$

you pick arms 10 times with the above policy.

Reward of an arm $\sim N(i, 1)$ ↳ arm number.

We have 5 odd arms & 5 even arms

$\therefore (p + p + p + p + p) + (2p + 2p + 2p + 2p + 2p) = 1$

$15p = 1$

$$\boxed{p = 1/15}$$

$$P[a_o] = \frac{1}{15} \qquad P[a_e] = \frac{2}{15}$$

At any time step, let's say you use the policy to pick an arm 'A_t', & receive R_t.

The Reward R_t is a R.V as policy is Stochastic. ∴ we need $E[R_t]$ at every t.

This we can calculate by summing over 'f' for all 10 arms. where

$$f_*^a = \left(\begin{matrix}\text{true mean} \\ \text{of the arm} \\ \text{'a'}\end{matrix}\right)\left(P[\text{choosing the arm}_{\text{'a'}}]\right)$$

at $t = 1$ $\quad f_t = E[R_t]$ basically

$$f_1 = (q^*(1))P[1] + (q^*(2))P[2] + \ldots (q^*_{[10]})P[10]$$

$$f_1 = 1p + 2(2p) + 3(p) + \ldots 10(2p)$$

$$f_1 = p(1+3+5+7+9) + 2p(2+4+6+8+10)$$

$$f_1 = \frac{25}{15} + \frac{60}{15}$$

$$f_1 = \frac{85}{15} = \frac{15}{3} \simeq 5.66$$

Since, our policy & true mean is the same for every time step. $\therefore$

$$f_1 = f_2 = f_3 \ldots\ldots f_{10}$$

$$E[R_1 + R_2 + \ldots R_{10}] = E[R_1] + E[R_2] + \ldots E[R_{10}]$$

$$E[\underbrace{R_1 + R_2 + R_3 + \ldots R_{10}}_{G_*}] = \underbrace{5.66 + 5.66 + \ldots}_{10 \text{ times}}$$

$$\boxed{E[G_*] = 56.66}$$

AUS

2)

$A = \{1, 2, 3 \ldots\ldots 10\}$

$B = \{1, 2, 4, 5, 7, 9, 10\}$

$C = \{3, 6, 8\}$

Dividing set of all arms into 2 subsets.

$$R_t(B) = \begin{cases} 0, & P[R_t(B) = 0] = 0.5 \\ 1, & P[R_t(B) = 1] = 0.5 \end{cases}$$

$$R_t(C) = \begin{cases} 0, & P[R_t(C) = 0] = 0.3 \\ 0.2, & P[R_t(C) = 0.2] = 0.3 \\ 1, & P[R_t(C) = 1] = 0.4 \end{cases}$$

Find Policies to maximize expected Reward.

$$E[R_t(B)] = 0(0.5) + 1(0.5) = 0.5$$

$$E[R_t(C)] = 0(0.3) + 0.2(0.3) + 1(0.4) = 0.46$$

$$E[R_t(B)] > E[R_t(C)]$$

∴ As long as our policy ensures that we pick an arm from the subset 'B'. We are good!

Our policies can be :-

1) only pick arms (1, 2, 4)

$P[1] = \frac{1}{3}$ $P[2] = \frac{1}{3}$ $P[4] = \frac{1}{3}$

rest zero.

2) only pick arm 1.

$P[1] = 1$, rest zero.

3) $P[$ picking 1st +ve multiple of 3$] = 0$

$P[$ picking 2nd +ve multiple of 3$] = 0$

$P[$ picking 4th +ve multiple of 2$] = 0$

Rest equally Distributed $\frac{1}{7}$ each.

4) $P[9] = \frac{1}{2}$ $P[7] = \frac{1}{4}$ $P[1] = \frac{1}{4}$

rest zero

5) $P[9] = 1$, rest zero

6) $P[9] = 0.8$ $P[1] = 0.2$, rest zero

3) Explore only non greedy Choices.

A = 3 arms : $\{a_1, a_2, a_3\}$

$$R_t(A) = \begin{cases} 0, & p = 1/2 \\ 1, & p = 1/2 \end{cases}$$

$Q_1(a)$ is the initial estimate (non zero)

$Q_1(a) = \{q_1, q_2, q_3\}$

$Q_1(a_1)$ $Q_1(a_2)$ $Q_1(a_3)$

$q_1 \neq q_2 \neq q_3 \neq 0$

Assuming $q_3 > q_2 > q_1$

Explain a sequence of $Q_t(a)$, A_t, R_t for $t = 1, 2, 3, 4, 5, 6$.

Explore at odd times.
Exploit at even times.

Ans

The time sequnce is following.

We start with initial Q : $Q_1(A)$. On the basis of this and our policy we make an Action A_1 & get a reward R_1 at $t = 1$. We will then calculate Q_2 for the arms & so on

$t = 1$, we explore (As per our policy)

As per our assumption $q_3 > q_2 > q_1$

∴ We can't Choose a_3 as we are exploring & we can't pick greedy Action while exploring in this question.

$$\therefore P[A_1 = a_1] = P[A_1 = a_2] = \frac{1}{2}$$

$$E[R_1] = \sum_r r \, P[r]$$

$$E[R_1] = 0\left(P[R_1 = 0]\right) + 1\left(P[R_1 = 1]\right)$$

$$E[R_1] = 0\Big(P[A_1 = a_1]\, P[R_1 = 0 \mid A_1 = a_1] + P[A_1 = a_2]\, P[R_1 = 0 \mid A_1 = a_2]\Big) + 1\Big(P[A_1 = a_1]\, P[R_1 = 1 \mid A_1 = a_1] + P[A_1 = a_2]\, P[R_1 = 1 \mid A_1 = a_2]\Big)$$

$$E[R_1] = 0 + 1\left(\left(\frac{1}{2} \times \frac{1}{2}\right) + \left(\frac{1}{2} \times \frac{1}{2}\right)\right)$$

$$\boxed{E[R_1] = \frac{1}{2}}$$

This is highly obvious as only rewards possible were 0 or 1 & both have equal p.

Hmmm, to continue we need to assume some stuff.

Let's say we explored and picked arm 2 'a_2'. Reward we got was 1.

$$A_1 = a_2 \qquad R_1 = 1$$

$(0, 1, 0)$

$$\therefore \quad Q_2(a_1) = q_1$$
$$Q_2(a_2) = q_2 + 1$$
$$Q_2(a_3) = q_3$$

$t = 2$

Now let's say $q_2 + 1$ ended q_3.

$\therefore$ a_2 is now the greedy choice.

We exploit at even times.

$$A_2 = a_2$$

Say, we get $R_2 = 0$ this time.

$E[R_2] = \frac{1}{2}$ still BTW. Even if we don't the if $Q_2(a_2) > Q_2(a_3)$, still $E[R_2] = 1/2$ as there are just 2

rewards {0, 1} & both have equal chance of being picked.

But sticking to our assumption

$$Q_2(a_2) > Q_2(a_3)$$

(0, 2, 0)

$\therefore$ $A_2 = a_2$, $R_2 = 0$ (let's say)

$$Q_3(a_1) = q_1$$

$$Q_3(a_2) = \frac{q_2 + 1 + 0}{2}$$

$$Q_3(a_3) = q_3$$

No more considering $E[R_*]$ as $E[R_t] = \frac{1}{2}$ always

$t = 3$

Let's say $Q_3(a_2)$ is the largest now

We have to explore now. say we pick arm a_1 by $\rho = 1/2$ and get the reward 1.

$\therefore$ $A_3 = a_1$, $R_3 = 1$

(1, 2, 0)

$$Q_4(a_1) = q_1 + 1$$

$$Q_4(a_2) = \frac{q_2 + 1}{2}$$

$$Q_4(a_3) = q_3$$

$t = 4$ (Exploit)

previous Assumption

Initially $q_3 > q_2 > q_1$

but at $t = 2$ we said

$q_2 + 1 > q_3 > q_1$

Now let's say

$Q_4(a_3)$ is the largest

(Doesn't contradict previous assumptions)

So, we pick a_3 & get $R_4 = 1$.

$$A_4 = a_3 \qquad R_4 = 1$$

(1, 2, 1)

This is the first time we picked a_3

$$Q_5(a_1) = (q_1 + 1)/1$$

$$Q_5(a_2) = \frac{q_2 + 1}{2}$$

$$Q_5(a_3) = \frac{q_3 + 1}{1}$$

$t = 5$

obviously if q_3 was the largest before. $q_3 + 1$ is the largest now.

we exploit & get :-

$A_5 = a_1$ (second time a_1 is picked)

$R_5 = 1$ (2, 2, 1)

$$Q_6(a_1) = \frac{q_1 + 1 + 0}{2} = \frac{q_1 + 1}{2}$$

$$Q_6(a_1) = \frac{q_1 + 1}{2}$$

$$Q_6(a_2) = \frac{q_2 + 1}{2}$$

$$Q_6(a_3) = q_3 + 1$$

$t = 6$

obviously $q_3 + 1$ is the largest & we exploit.

$R_6 = 1 \qquad A_6 = a_3$

$$Q_7(a_1) = \frac{q_1 + 1}{2}$$

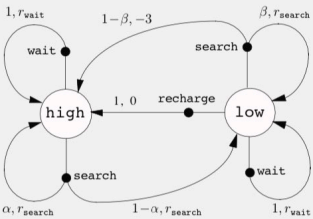
$$Q_7(a_2) = \frac{q_2 + 1}{2}$$

$$Q_7(a_3) = \frac{q_3 + 2}{2}$$

A mobile robot has the job of collecting empty soda cans in an office environment. It has sensors for detecting cans, and an arm and gripper that can pick them up and place them in an onboard bin; it runs on a rechargeable battery. The robot's control system has components for interpreting sensory information, for navigating, and for controlling the arm and gripper. High-level decisions about how to search for cans are made by a reinforcement learning agent based on the current charge level of the battery. To make a simple example, we assume that only two charge levels can be distinguished, comprising a small state set $\mathcal{S} = \{\texttt{high}, \texttt{low}\}$. In each state, the agent can decide whether to (1) actively `search` for a can for a certain period of time, (2) remain stationary and `wait` for someone to bring it a can, or (3) head back to its home base to `recharge` its battery. When the energy level is `high`, recharging would always be foolish, so we do not include it in the action set for this state. The action sets are then $\mathcal{A}(\texttt{high}) = \{\texttt{search}, \texttt{wait}\}$ and $\mathcal{A}(\texttt{low}) = \{\texttt{search}, \texttt{wait}, \texttt{recharge}\}$.

The rewards are zero most of the time, but become positive when the robot secures an empty can, or large and negative if the battery runs all the way down. The best way to find cans is to actively search for them, but this runs down the robot's battery, whereas waiting does not. Whenever the robot is searching, the possibility exists that its battery will become depleted. In this case the robot must shut down and wait to be rescued (producing a low reward). If the energy level is `high`, then a period of active search can always be completed without risk of depleting the battery. A period of searching that begins with a `high` energy level leaves the energy level `high` with probability α and reduces it to `low` with probability $1 - \alpha$. On the other hand, a period of searching undertaken when the energy level is `low` leaves it `low` with probability β and depletes the battery with probability $1 - \beta$. In the latter case, the robot must be rescued, and the battery is then recharged back to `high`. Each can collected by the robot counts as a unit reward, whereas a reward of -3 results whenever the robot has to be rescued. Let $r_{\texttt{search}}$ and $r_{\texttt{wait}}$, with $r_{\texttt{search}} > r_{\texttt{wait}}$, denote the expected number of cans the robot will collect (and hence the expected reward) while searching and while waiting respectively. Finally, suppose that no cans can be collected during a run home for recharging, and that no cans can be collected on a step in which the battery is depleted. This system is then a finite MDP, and we can write down the transition probabilities and the expected rewards, with dynamics as indicated in the table on the left:

s	a	s'	$p(s'\|s,a)$	$r(s,a,s')$
high	search	high	α	$r_{\texttt{search}}$
high	search	low	$1-\alpha$	$r_{\texttt{search}}$
low	search	high	$1-\beta$	-3
low	search	low	β	$r_{\texttt{search}}$
high	wait	high	1	$r_{\texttt{wait}}$
high	wait	low	0	-
low	wait	high	0	-
low	wait	low	1	$r_{\texttt{wait}}$
low	recharge	high	1	0
low	recharge	low	0	-



Note that there is a row in the table for each possible combination of current state, s, action, $a \in \mathcal{A}(s)$, and next state, s'. Some transitions have zero probability of occurring, so no expected reward is specified for them. Shown on the right is another useful way of

summarizing the dynamics of a finite MDP, as a *transition graph*. There are two kinds of nodes: *state nodes* and *action nodes*. There is a state node for each possible state (a large open circle labeled by the name of the state), and an action node for each state–action pair (a small solid circle labeled by the name of the action and connected by a line to the state node). Starting in state s and taking action a moves you along the line from state node s to action node (s, a). Then the environment responds with a transition to the next state's node via one of the arrows leaving action node (s, a). Each arrow corresponds to a triple (s, s', a), where s' is the next state, and we label the arrow with the transition probability, $p(s'|s, a)$, and the expected reward for that transition, $r(s, a, s')$. Note that the transition probabilities labeling the arrows leaving an action node always sum to 1.

4)

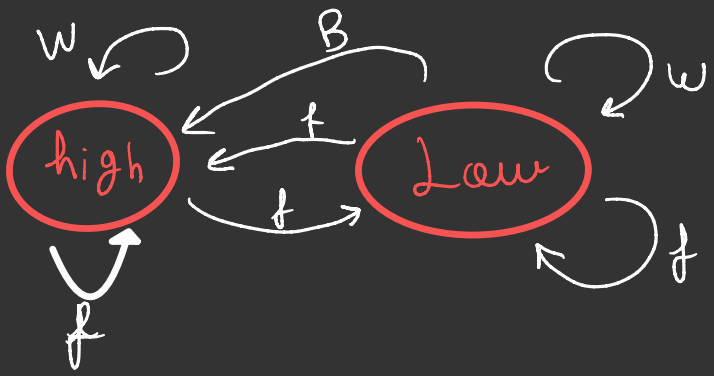
$S = \{high, low\}$

$A = \{search, wait, recharge\}$

$A(high) = \{search, wait\}$

$A(low) = \{search, wait, recharge\}$

If battery depleted then rescue.



Search → f
Wait → W
Recharge → B

* $P[S_{t+1} = h \mid h, w] = 1$
* $P[S_{t+1} = h \mid l, B] = 1$
* $P[S_{t+1} = h \mid h, f] = \alpha$
* $P[S_{t+1} = l \mid h, f] = 1-\alpha$
* $P[S_{t+1} = h \mid l, f] = 1-\beta$ (Battery depletes, so it gets rescued ∴ high energy rent)
* $P[S_{t+1} = l \mid l, f] = \beta$
* $P[S_{t+1} = l \mid l, w] = 1$

	$S_t = s$ $\{H, L\}$	$S_{t+1} = s'$ $\{H, L\}$	$A_t = a$ $a \in \{f, w, B\}$	$P[s' \mid s, a]$	$R_t = r$ read the table for reward its set.
1)	H	H	f	α	r_f
2)	H	L	f	$1-\alpha$	r_f
3)	H	H	w	1	r_w
4)	H	L	w	0	–
5)	L	L	f	β	r_f
6)	L	L	w	1	r_w
7)	L	H	f	$1-\beta$	-3
8)	L	H	w	0	–
9)	L	H	B	1	0
10)	L	L	B	0	–

reward for entry 7 is not $r_f - 3$, because its given no robot can be collected if a step taken results in battery Depletion

s	s'	a	r	$P[s', r \mid s, a]$
H	H	w	0	1
H	H	f	r_f	α
H	H	f	$r \neq r_f$	0
H	H	w	$r = r_w$ $r \neq r_w$	! 0
H	L	f	r_f	$1-\alpha$
H	L	f	$r \neq r_f$	0
H	L	w	any 'r'	0
H	L	B	any 'r'	0
L	H	f	$r = -3$	$1-\beta$
L	H	f	$r \neq -3$	0
L	H	w	any 'r'	0
L	H	B	$r = 0$ $r \neq 0$	1 0
L	L	B	any 'r'	0
L	L	w	r_w $r \neq r_w$	1 0
L	L	f	$r = r_f$	β
L	L	f	$r \neq r_f$	0

8) u_* in terms of q_*

$$u_*(s) = \max_{a \in A} \left(E\left[R_{t+1} + \gamma u_*(S_{t+1}) \mid S_t = s, A_t = a \right] \right)$$

$$\boxed{u_*(s) = \max_{a \in A} q_*(s)}$$

Ans

11)

R_{t+1} depends on S_t, A_t. But does R_{t+2} depend on S_t, A_t?

$$P[R_{t+2} = r'' \mid S_t = s, A_t = a]$$

$$P[r'' \mid s, a] = \sum_{a'} \sum_{s'} P[r'', s', a' \mid s, a]$$

$$P[r'' \mid s, a] = \sum_{a'} \sum_{s'} \Big(P[r'' \mid s, a, s', a'] \; P[a', s' \mid s, a] \Big)$$

↓

Here if we know s', a' then we don't have to condition upon (s, a)

$$P[r'' \mid s, a] = \sum_{a'} \sum_{s'} P[r'' \mid s', a'] P[s', a' \mid s, a]$$

↓

MDP form

more generally :-

$$P[r'' \mid s, a] = \sum_{s''} \sum_{a'} \sum_{s'} P[r'', s'' \mid s', a'] \; P[s', a' \mid s, a]$$

12)

$$E[R_{t+2} \mid S_t = s, A_t = a] = \sum_{r'' \in R(s'')} r'' P[r'' \mid s, a]$$

$$= \sum_{r'' \in R(s'')} r'' \sum_{s''} \sum_{a'} \sum_{s'} P[r'', s'' \mid s', a'] P[s', a' \mid s, a]$$

---

13) $v_\pi(s) = E[G_t \mid S_t = s]$

$$v_\pi(s) = E[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots \mid S_t = s]$$

$$v_\pi(s) = E[R_{t+1} + \gamma(R_{t+2} + \gamma R_{t+3} + \cdots) \mid S_t = s]$$

$$v_\pi(s) = E[R_{t+1} + \gamma G_{t+1} \mid S_t = s]$$

$$v_\pi(s) = E[R_{t+1} \mid S_t = s] + \gamma E[G_{t+1} \mid S_t = s]$$

from iterated expectations

$$v_\pi(s) = E[R_{t+1} \mid S_t = s] + \gamma E[v_\pi(S_{t+1}) \mid S_t = s]$$

$$v_\pi(s) = E[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s]$$

$$v_\pi(s) = \sum_r \sum_{s'} (r + \gamma v_\pi(s')) P[R_{t+1} = r, S_{t+1} = s' \mid S_t = s]$$

$$v_\pi(s') = \sum_a \sum_r \sum_{s'} (r + \gamma v_\pi(s')) \, p[r, s', a \mid s]$$

$$v_\pi(s') = \sum_a \sum_r \sum_{s'} \left[(\pi(a \mid s)) \, (r + \gamma v_\pi(s')) \, (p[r, s' \mid s, a]) \right]$$

$$\boxed{v_\pi(s') = \sum_a \pi[a \mid s] \sum_r \sum_{s'} (r + \gamma v_\pi(s')) \, p[r, s' \mid s, a]}$$

14) $R_1 = 2$ $R_2 = -1$ $R_3 = 10$

$R_4 = -3$ $\gamma = 0.5$

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + \cdots$$

$\therefore G_4 = G_5 = G_6 = G_7 \cdots = 0$

* $G_3 = R_4 + \gamma R_5 + \cdots$

$G_3 = -3$

*

$$G_2 = R_3 + \gamma R_4 = 10 + \frac{1}{2}(-3)$$

$G_2 = 8.5$

Alternatively

$G_2 = R_3 + \gamma G_3$

$G_2 = 10 + \frac{1}{2}(-3) = 8.5$

*

$$G_1 = R_2 + \gamma R_3 + \gamma^2 R_4$$

$$G_1 = -1 + \frac{10}{2} + \left(-\frac{3}{4}\right) = \frac{-4 + 20 - 3}{4}$$

$G_1 = 3.25$

Alternatively

$G_1 = R_2 + \gamma G_3$

$G_1 = -1 + \frac{8.5}{2} = 3.25$

*

$$G_0 = R_1 + r R_2 + r^2 R_3 + r^3 R_4$$

$$G_0 = 2 + -\frac{1}{2} + \frac{10}{4} + -\frac{3}{8} = 3.625$$

$$G_0 = R_1 + r G_1$$

$$G_0 = 2 + \frac{3.25}{2} = \frac{7.25}{2}$$

$$\boxed{G_0 = 3.625}$$

* $R_t = C \quad \forall \quad t$

$$\therefore G_t = C + rC + r^2 C + r^3 C + \cdots$$

$$G_t = C(1 + r + r^2 + r^3 + \cdots)$$

$r < 1 \Rightarrow \infty$ GP which converges

$$G_t = C\left(\frac{1}{1-r}\right) = \frac{C}{1-r}$$

$$\boxed{G_t = \frac{C}{1-r}}$$

↳ Sum of infinite GP w/t $r < 1$: $\frac{a}{1-r}$

15) OPTIMAL POLICY FROM OAVF

Given : $q_*(s,a)$ $\forall$ (s,a) pairs $s \in S$, $a \in A(s)$

$$\pi_*(s) = \underset{a \in A(s)}{\operatorname{argmax}} \left(q_*(s,a) \right)$$

From $\pi_* \longrightarrow q_{\pi_*}$ or v_{π_*}

write the Bellman Equation corresponding to π_* & solve them for both v_{π_*} & q_{π_*}.

Later in Course

15) OPTIMAL POLICY FROM OSVF

Given: $v_*(s,a)$ $\forall s \in S$, Find $\pi^*(s)$

$\pi^*(s)$

$$\underset{a \in A(s)}{\operatorname{argmax}}\ q_*(s,a) = \underset{a \in A(s)}{\operatorname{argmax}}\ \mathbb{E}\left[R_{t+1} + \gamma v_*(S_{t+1}) \,\middle|\, S_t = s, A_t = a \right]$$

$$\underset{a \in A(s)}{\operatorname{argmax}}\ q_*(s,a) = \underset{a \in A(s)}{\operatorname{argmax}}\ \sum_{r} \sum_{s'} (r + \gamma v_*(s')) P(s', r \mid s, a)$$

Calculate this sum for all $a \in A(s)$ & choose a which maximizes it.