Birla Institute of Technology & Science, Pilani
Work Integrated Learning Programmes Division
First Semester 2024-2025

Mid-Semester Test
(EC-2 Makeup)

Course No.	: AIMLCZG512
Course Title	: Deep Reinforcement Learning
Nature of Exam	: Closed Book
Weightage	: 30%
Duration	: 2 Hours
Date of Exam	: _________ EN

No. of Pages = 3
No. of Questions = 5

Note to Students:

1. Please follow all the *Instructions to Candidates* given on the cover page of the answer book.
2. All parts of a question should be answered consecutively. Each answer should start from a fresh page.
3. Assumptions made if any, should be stated clearly at the beginning of your answer.

Q.1. [2+2+2=6 Marks]

In social media marketing, influencers (seeds) are given free products or discounts to spread product awareness through word-of-mouth in their networks. Marketers measure success by the number of shares or likes. This process can be modelled as a multi-armed bandit (MAB) problem, where actions involve selecting influencers, and rewards depend on their influence metric.

The action values are initialised to 0, to begin with. Below are the results from 8 time steps of MAB action:

Time Step	Influencer ID	Influence (No. of. Shares/Likes)
1	3	30
2	5	50
3	2	60
4	4	50
5	3	50
6	5	50
7	5	20
8	2	50

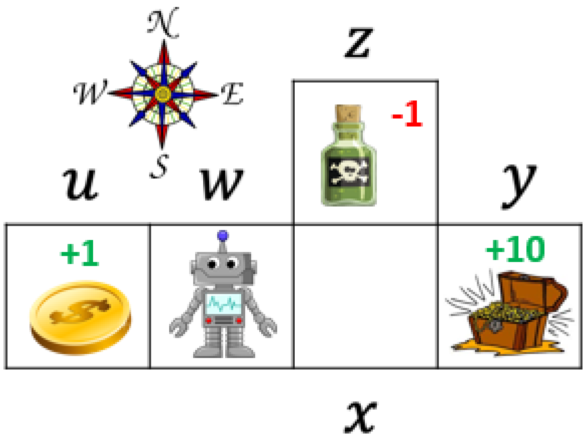
(a) Identify the next two actions using UCB action selection. Assume c to be 0.7. [2.0 M].

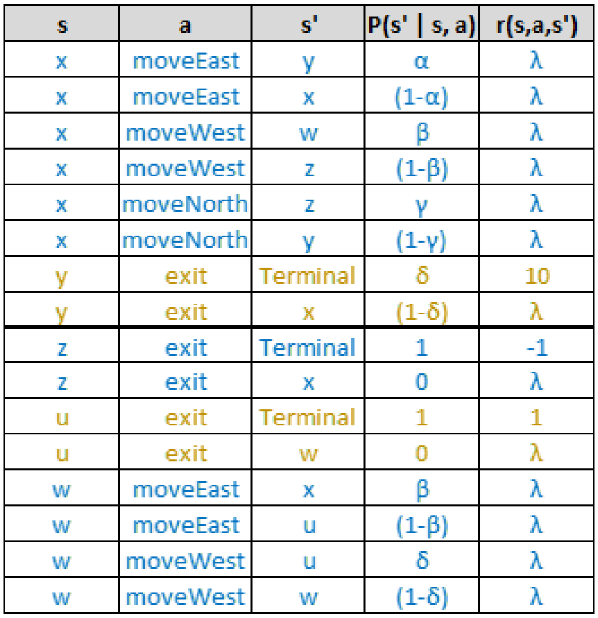
(b) How would you choose the next two actions using ε-greedy? Explain. Assume ε = 0.7. [2.0 M].

(c) Assume the reward distribution varies with time as it is valid for such a problem. Re-estimate the value of Influencers with IDs 3 and 5. [Note: assume $\alpha = 0.5$, if required] [2.0 M].

Q.2. [2+2+1+1=6 Marks]

Recollect the treasure-hunting navigation robot example discussed in the class as depicted below. Consider the problem of designing a reinforcement-learning agent for the bot whose dynamics is reproduced below.





s	a	s'	P(s' | s, a)	r(s,a,s')
x	moveEast	y	α	λ
x	moveEast	x	(1-α)	λ
x	moveWest	w	β	λ
x	moveWest	z	(1-β)	λ
x	moveNorth	z	γ	λ
x	moveNorth	y	(1-γ)	λ
y	exit	Terminal	δ	10
y	exit	x	(1-δ)	λ
z	exit	Terminal	1	-1
z	exit	x	0	λ
u	exit	Terminal	1	1
u	exit	w	0	λ
w	moveEast	x	β	λ
w	moveEast	u	(1-β)	λ
w	moveWest	u	δ	λ
w	moveWest	w	(1-δ)	λ

i) Suggest any one valid set of values for (α, β, γ, δ) w.r.t each of the below cases and justify your choice in no more than one sentence: [2.0 M].

a) If the robot works in a deterministic environment

b) If the robot's action outcomes are stochastic in nature

ii) Write down the bellman expressions for $v_*(w)$, $q_*(x, \text{moveEast})$ that suit the given dynamics, i.e. make the bellman equations to be very specific to this case [2.0 M].

iii) Illustrate the significance of discounting using the above part ii) answer for any one of the expressions. Explain in no more than two sentences and be specific to the given use case. [1.0 M].

iv) Comment on the robot's behaviour if λ is zero. [1.0 M].

Q3. [1+1+1+1=4 Marks]

A robot navigates a 3x3 grid to reach a target location. The robot starts at position (0, 0), and the target is at position (2,2). The robot can move up, down, left, or right, with a reward of −1 for each move. The task is episodic, meaning the robot restarts after reaching the target or running out of steps. If the robot reaches the target, it gets a reward of +10. For the robot's task in the given scenario, define the following components of the Markov Decision Process (MDP):

i. State Space: List and explain the states the robot can be in. [1.0 M].

ii. Action Space: List the actions available to the robot in each state. [1.0 M].

iii. Reward Function: Define the rewards for each state-action pair. [1.0 M].

iv. State-Transition Probabilities: Describe the transition probabilities for the robot moving in the grid, assuming no stochasticity (i.e., deterministic transitions). [1.0 M].

Q4. [2+3=5 Marks]

A drone is tasked with delivering packages to four locations arranged by increasing altitude: low, medium, high, and top. The drone starts at the low altitude and must reach the top while minimizing its energy consumption. It can operate in two flight modes. In the low-power flight mode, the drone moves one level up (e.g., from low to medium, medium to high, or high to top) with a 20% probability. However, there is a 80% probability of falling back to the low altitude, regardless of its current position. This mode consumes 1 energy unit per step. In the high-power flight mode, the drone moves one level up with a 60% probability, but there is also a 40% probability of falling back to the low altitude. This mode consumes 2 energy units per step. [Note: make and state assumptions, only if required]

(a) Describe dynamics in Markov Decision Process (MDP) using a neat table for this scenario, considering the states (low, medium, high, and top), actions (low-power and high-power modes), transition probabilities, and energy costs. [2.0 M].

(b) Perform one iteration of value iteration [use asynchronous update] using a discount factor of 0.6, starting with all state values initialised to 0. Calculate the updated values for all states and determine whether the results suggest any changes to the drone's optimal policy. [3.0 M].

Q5. [3+1=4 Marks]

In off-policy learning, learning the agent involves using a behaviour policy b and a target policy π. Here, the target policy π is deterministic. The agent can be in states S1 or S2 at any point in time. The actions Up and Down are available from all the states. The behavior and target policies are as below:

Behaviour Policy b	Up	Down
S1	0.2	0.8
S2	0.8	0.2

Target Policy π	Up	Down
S1	1	0
S2	0	1

Consider the following episode generated by the behaviour policy:

S1 - Up - (+2) - S2 - Down - (+4) - S1 - Down - (+4) - S2 - Down - (+8) - S1 - Down - (+8) - S2 - Down - (+8) - S1 - Up - (+8) - S2 - Down - (+8)

(i) Estimate the value of the state-value (S2, Down) and (S2, Up) using first visit estimate. Show all the steps. Assume the discount factor to be 0.9. [3 Marks]

(ii) Assume the target policy π for S2 is updated after the estimation in (a). Show the target policy for S2. [1 Marks]

Q6 [1+1+3=5 Marks]

(a) Write two issues when the behaviour policy used in Off-policy Monte Carlo learning is deterministic.[Only 2 points required, [1 Marks]

(b) In the Monte Carlo method, the learning of values/policy is done using the episodes. How do we ensure the episodes generated for learning policies encourage exploration? Give one point in support or On-Policy Learning and one point in support of Off-Policy Learning. [1 Marks]

(c) Write down either policy iteration or value iteration algorithm. How do you describe its complexity? Write two comments about the learned policy. [3 Marks]

****

SOLUTION

① (a) Next 2 actions to be found using UCB action selection.

- Assume $c = 0.7$.

* Calculate avg reward $Q(a)$ & No. of times $N(a)$ each influencer was selected.

ID 2 : $N = 2$, Avg $= (60+50)/2 = 55$

ID 3 : $N = 2$, Avg $= (30+50)/2 = 40$

ID4 : $N = 1$, Avg $= 50$

ID5 : $N = 3$, Avg $= (50+50+20)/3 \approx 40$

* Compute UCB for each influencer.

ID 2 : $55 + 0.7\sqrt{\ln(8))/2} \approx 55 + 0.7(0.739) \approx 55.517$

ID 3 : $40 + 0.7\sqrt{\ln(8))/2} = 40 + 0.7(0.739) = 40.517$

ID4 : $50 + 0.7\sqrt{\ln(8))/1} = 50 + 0.7(1.044) = 50.731$

ID5 : $40 + 0.7\sqrt{\ln(8))/3} = 40 + 0.7(0.603) = 40.422$

Next 2 actions (highest UCB) = $\boxed{\text{ID2 \& ID4}}$

(b) Next 2 actions using ϵ-greedy. $\epsilon = 0.7$

$\epsilon = 0.7 \Rightarrow$ 70% explore (random)

30% exploit (choose one with high average reward)

From average,

Best ID : ID2 (55) then ID4 (50)

70% of time pick randomly

30% of time choose ID2 first, then ID4.

(c) Reward distribution varies with time.

Reestimate the value of influencers with IDs 3 & 5.

Assume $\alpha = 0.5$

* We can use the incremental reward computation formula

$$Q_{n+1} = Q_n + \alpha (R_n - Q_n)$$

↳ Avg reward so far.

ID3 : $Q = 40$, new reward $= 50$

$$Q' = 40 + 0.5 (50 - 40) = 45$$

ID5 : $Q = 40$, new reward $= 20$

$$Q' = 40 + 0.5 (20 - 40) = 30.$$

2 (i) Suggest valid set of values for $\alpha, \beta, \gamma, \delta$

(a) If robot works in deterministic env.

$\alpha = 1$, $\beta = 1$, $\gamma = 1$, $\delta = 1$

When it is deterministic, the next state mentioned is deterministic with prob. 1.

(b) Stochastic

The env. outcome is random

$\alpha = 0.3$[redacted], $\beta = 0.3$, $\gamma = 0.3$, $\delta = 0.5$ $\boxed{1-\delta = 0.5}$

$\boxed{1-\beta = 0.7}$ $\boxed{1-\gamma = 0.7}$

$\boxed{1-\alpha = 0.7}$

[From x, 3 options are there - w, z, y with 33.3% each]

(ii) Bellman expression

$$v^*(w) = \max_a \sum_{s'} P(s' | w, a) [R + \gamma v^*(s')]$$

$$q^*(x, \text{move East}) = \sum_{s'} P(s' | x, \text{moveEast}) [R + \gamma \max_{a'} q^*(s', a')]$$

(iii) Discounting significance

For q^* (x, moveEast), discounting ensures the robot prefers faster paths to the goal. A lower γ would discourage long-term goals.

(iv) $\lambda = 0$

No importance to future rewards.

Slow learning.

③ (i) State space

9 states : 0×0, 0×1, 0×2
1×0, 1×1, 1×2
2×0, 2×1, 2×2

(ii) Action space : {up, down, left, right}

(iii) Reward function : Each move = −1
Reaching (2×2) = +10

(iv) State transition probability [Deterministic]

$P(s' | s, a) = 1$ for valid move

④ (a) MDP table

State	Action	Next State	Prob.	Reward (Energy cost)
Low	Low power (LP)	Medium	0.2	−1
Low	Low power	Low	0.8	−1
Low	high power (HP)	Medium	0.6	−2
Low	HP	Low	0.4	−2
Medium	LP	High	0.2	−1
Medium	LP	Low	0.8	−1
Medium	HP	High	0.6	−2
Medium	HP	Low	0.4	−2

High	LP	Top	0.2	-1	
High	LP	Low	0.8	-1	
High	HP	Top	0.6	-2	
High	HP	Low	0.4	-2	
Top	-	Top	1.0	0	(Terminal State)

(b) Value iteration

$\gamma = 0.6$, initial values = 0.

Bellman optimality update formula for value iteration is

$$V(s) \leftarrow \max_a \sum_{s'} P(s' \mid s,a)\,[R(s,a,s') + \gamma \cdot V(s')]$$

Ex: $V_{LP}(Low) = 0.2(-1 + 0.6 \cdot V(Medium)) + 0.8\,[-1 + 0.6 \cdot V(Low)]$

$V = 0$ [initial]

$= 0.2(-1) + 0.8(-1) = -1.$

$V_{HP}(Low) = 0.6(-2 + 0.6 \cdot V(Medium)) + 0.4(-2 + 0.6 \cdot V(Low))$

$= 0.6(-2) + 0.4(-2\cancel{4}) = -2$

$V(Low) = \max(-1, -2) = -1.$

Similarly, we have to compute for Medium & high.

5 Episode

S1-up (+2) - S2-down (+4) - S1-down (+4) - S2-down (+8) - S1-down (+8), S2-down (+8), S1-up (+8), S2-down (+8)

(i) First value estimate, $\gamma = 0.9$

$(S2, down) = 4 + 0.9(4) + 0.9^2(8) + 0.9^3(8) + 0.9^4(8)$
$+ 0.9^5(8) \approx 34.63$

(S2, up) = Not taken in this episode.

(ii) Since (s2, down) has high return

π (s2) = Down. [policy].

⑥ (a) 2 issues when behavior policy used in off-policy monte Carlo learning is deterministic.

(i) No exploration – Can't learn from unseen states/actions.

(ii) Bias – target policy not fully evaluated.

(b) Exploration encouragement

On-policy : uses ϵ-greedy (ensures random exploration)

Off-policy : Behavior policy is different (allows diverse episodes)

(c) policy iteration or value iteration algorithm (direct question).