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

Comprehensive Exam
(EC-3 Make-up)

Q1) [3+5 = 8 Marks]

(a) A company is running an A/B/C/D test on four different versions of its website to determine which design gets the highest user engagement (measured as a click-through rate (CTR)). They model this as a multi-armed bandit problem, where each version corresponds to an arm of the bandit. They decide to use the ϵ-greedy strategy with an initial exploration rate of ϵ=0.15, which decays over time. At a particular step in the experiment, the estimated click-through rates (Q-values) for the four versions are:

Q1 = 0.05, Q2 = 0.12, Q3 = 0.09, Q4 = 0.15

(a-1) Explain how the ϵ-greedy strategy balances exploration and exploitation. [0.5 M]
(a-2) What is the effect of using a decaying ϵ rather than a fixed ϵ? [0.5 M]
(a-3) Compute the probability of selecting each version of the website under the ϵ=0.15-greedy strategy. [1 M]
(a-4) Compute the expected reward at this step, given the true CTRs for each website version and the action probabilities. R1 = 0.08, R2 = 0.15, R3 = 0.10, R4 = 0.18. [1 M]

Answer:
(a-1) The ϵ-greedy strategy balances exploration and exploitation by choosing a random action (exploration) with probability ϵ, and the best-known action (exploitation) with probability 1 - ϵ. This allows the algorithm to explore potentially better options while mostly favoring the option currently believed to be the best. [0.5 Marks]

(a-2) Using a decaying ϵ allows the algorithm to explore more in the early stages (when knowledge about the environment is limited) and gradually shift towards exploitation as it gains more confidence in its estimates.

(a-3)Best arm (highest Q-value) is Q4 = 0.15
With ϵ = 0.15:
Exploitation: choose Q4 with probability 1 - ϵ = 0.85
Exploration: choose randomly among all 4 arms with probability ϵ = 0.15, so each arm gets 0.15 / 4 = 0.0375
So the probabilities:

P1 = 0.0375
P2 = 0.0375
P3 = 0.0375
P4 = 0.85 + 0.0375 = 0.8875 [1 Mark]

(a−4) Expected reward E[R] = P1·R1+P2·R2+P3·R3+P4·R4
= (0.0375·0.08) + (0.0375·0.15) + (0.0375·0.10) + (0.8875·0.18)
= 0.172 [1 Mark]

Additional Note to TA on Marking Scheme:
(a-1) Clear mention of use of ϵ [0.5 Marks]
(a-2) Significance of decaying ϵ [0.5 Marks]
(a-3) Correct answer [1 Mark], incorrect answer and if steps are correct [0.5 Marks]
(a-4) Correct answer [1 Mark], incorrect answer and if steps are correct [0.5 Marks]

(b) A delivery robot operates in a 4x4 grid world. The robot can move up, down, left, or right and earns rewards based on the location it reaches. The environment is modelled as a Markov Decision Process (MDP) with the following characteristics:

State (s)	The robot's current grid position
Actions (a)	{Up, Down, Left, Right}

Transition Probability (P(s′|s,a))	80% chance the robot moves in the intended direction, 10% chance it deviates left, and 10% chance it deviates right
Rewards (R(s,a))	+10 for reaching the goal state (4,4). −5 for hitting an obstacle. 0 for all other moves. Discount Factor (γ): 0.9.

(b-1) Explain why this problem satisfies the Markov Property. [1 M]
(b-2) Assume the robot is in state (3,3) and follows an optimal policy. Using the Bellman equation, Show how to compute V(3,3) given the rewards and discount factor γ=0.9. [1.0 M]
(b-3) Assume the estimated rewards of states around (3,3) are are:
V(4,3) = 8.0, V(3,4) = 9.0, V(2,3) = 5.0, V(3,2) = 4.5
Compute V(3,3) using Value Iteration. [3 M]

Answers:
(b-1) This problem satisfies the Markov Property because the transition probabilities and rewards depend only on the current state and action, not on the sequence of states or actions that led to it. That is:
P(s′|s,a)=dependent only on s and a [1 Mark]

(b-2) The Bellman equation for a state V(s) is:

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

In this case, for state (3,3) with four possible actions, we compute expected value for each action and choose the one with the maximum expected value. [1 Mark]

(b-3) Action: Up

- Intended: (2,3) → V = 5.0
- Left of Up = Left = (3,2) → V = 4.5
- Right of Up = Right = (3,4) → V = 9.0

Q(Up) = 0.8*5.0 + 0.1*4.5 + 0.1*9.0 = 4.0 + 0.45 + 0.9 = 5.35

Action: Down

- Intended: (4,3) → V = 8.0
- Left of Down = Right = (3,4) → V = 9.0
- Right of Down = Left = (3,2) → V = 4.5

Q(Down) = 0.8*8.0 + 0.1*9.0 + 0.1*4.5 = 6.4 + 0.9 + 0.45 = 7.75

Action: Left

- Intended: (3,2) → V = 4.5
- Left of Left = Down = (4,3) → V = 8.0
- Right of Left = Up = (2,3) → V = 5.0

Q(Left) = 0.8*4.5 + 0.1*8.0 + 0.1*5.0 = 3.6 + 0.8 + 0.5 = 4.9

Action: Right

- Intended: (3,4) → V = 9.0
- Left of Right = Up = (2,3) → V = 5.0
- Right of Right = Down = (4,3) → V = 8.0

Q(Right) = 0.8*9.0 + 0.1*5.0 + 0.1*8.0 = 7.2 + 0.5 + 0.8 = 8.5

Apply the Bellman update:
V(3,3)=max$_{\{Up,Down,Left,Right\}}${Q(Up),Q(Down),Q(Left),Q(Right)}
V(3,3)=max$_{\{Up,Down,Left,Right\}}${5.35,7.75,4.9,8.5}=8.5 [3 Marks]

Additional Note to TA on Marking Scheme:
(b-1) Description of condition [1 Mark]
(b-2) Bellman equation and explanation [1 Mark]
(b-3) Q value calculation - 0.5 Marks each, Final answer 1 mark. Step marks can be given if final answer is alone wrong. [3 Marks]

Q2) [4.0+3.0 = 7 Marks]

Use case: An RL agent built to act as a team manager in the IPL (Indian Premier League 2025, is tasked with selecting players during IPL auction.
For simplicity, assume that the auction goes through only three rounds (round 1, round 2, round 3) in phase 1. In each round, one player is presented. RL agent assesses the suitability of the player and with equal likelihood can either choose the player for its team for a cost = – k units (where unit = 100 crores in monetary value and "k" correspond to the kth round of the auction) or choose to wait for the next player in the (k+1) round.
Obviously as the auction round moves towards the closure, there is a risk to settle with costlier players or end up with selecting none as only three rounds are organized. The agent can select a minimum of zero to maximum of three players at the end. If no player is selected, then on an average the bidder needs to spend -2 units on training the existing pool of players or use the same amount for purchasing in the next phase 2 auction.
As the agent goes along the auction, it does not know what the next player will bring in terms of skill! Also once an agent passes (ie., not selecting) on a player, it cannot come back to reselect them.

(a) Formulate the above given problem as a finite MDP. Depict the state transition diagram with clear labelling of reward , transition probabilities and terminal state [4.0 M]
(b) Using the results of part a) answer the following:
Assume discount factor = 1 and V(S) = -1 for all the states, perform only one iteration of generalized policy iteration (ie., One iteration of policy evaluation immediately followed by one iteration of policy improvement). [3.0 M]

Answer:

(a)

(b)

Policy Eval	Round 1 S_{r1}	Round 2 S_{r2}	Round 3 S_{r3}	$S_{Terminal}$
Initial Value	-1	-1	-1	
Value(S_{ri}) Bell man = $\sum_a \pi(a\|s) \sum_{s',r} p(s',r\|s,a)\left[r+\gamma v_\pi(s')\right]$	0.5 [(-1+1(-1)) + (0+1(-1))] = -1.5	0.5 [(-2+1(-1)) + (0+1(-1))] = -2	0.5 [(-3+1(**-2**)) + (0+1(**-2**))] = -3.5	The value -2 incurred as possible state value must be used to generalize here for V($S_{Terminal}$)
Alternate key for partial answers →			0.5[(-3+1(-1)) + (0+1(-1))] = -2.5	If above is not considered and -1 is uses as is partial marks can be awarded

Policy Iteration	Round 1 S_{r1}	Round 2 S_{r2}	Round 3 S_{r3}	
Update Policy ie., Best action	MAX { 0.5(-1+1(**-2**)) , 0.5(0+1(**-2**))} = MAX {-1.5, -1}	MAX { 0.5(-2+1(**-3.5**)) , 0.5(0+1(**-3.5**))} = MAX {-2.75, -1.75}	MAX { 0.5(-3+1(**-2**)) , 0.5(0+1(**-2**))} = MAX {-2.5, -1}	
Best Action	"Don't Select Player"	"Don't Select Player"	"Don't Select Player"	
		Round 2 : MAX { 0.5(-2+1(**-2.5**)) , 0.5(0+1(**-2.5**))} = MAX {-2.25, -1.25}	**Round 3** MAX { 0.5(-3+1(**-1**)) , 0.5(0+1(**-1**))} = MAX {-2, -0.5}	**←(for previous assumption of -1 instead of -2 consider for partial marking. Still the action result will be the same)**

Additional note to TA on Marking Scheme:

(a) State transition with three states identified = 1 mark
Probability of all transitions identified = 1 mark
Actions identified = 1 mark
Correct rewards per state identified = 1 mark

(b) Policy Evaluation step with use of bellman equation = 1.5 marks (0.5 marks per state)
Policy Iteration step with of MAX function to find the optimal action = 1.5 marks (0.5 marks per state)
If the answers are incorrect ie., if the students have assumed -1 for V(Terminal state) then a total 1.5 marks can be given as partial for part (b). Refer to the alternate answers in this case in above table's last row)

Q3) (a) [1+2+2+2= 7 Marks]

Consider an episodic reinforcement learning environment with three states, S = $\{s_1, s_2, s_3\}$, and two actions, A = $\{a_1, a_2\}$. Terminal state is s_3 and the transition details are depicted as below:

From s_1:
a_1, leads to s_2, with reward R = 1
a_2 leads to s_3, with reward R = 0.
-From s_2:
a_1, leads to s_3, with reward R = 2
a_2 leads to s_3, with reward R = 0.

We aim to estimate the value of state s_1 under a target policy π using off-policy Monte Carlo prediction with importance sampling, based on episodes generated by a behavior policy b. The policies are:

Target policy π:
$\pi(a_1 | s_1) = 0.7, \quad \pi(a_2 | s_1) = 0.3$
$\pi(a_1 | s_2) = 0.9, \quad \pi(a_2 | s_2) = 0.1$

Behavior policy b:
$b(a_1 | s_1) = 0.4, \quad b(a_2 | s_1) = 0.6$
$b(a_1 | s_2) = 0.5, \quad b(a_2 | s_2) = 0.5$

Suppose you have collected two episodes:
Episode 1: < s1, a1, R = 1, s2, a1, R = 2, s3 (terminal) >.
Episode 2: < s1, a2, R = 0, s3 (terminal) >

Answer the following given this scenario:
(a-1) Explain why importance sampling is required for off-policy prediction in this scenario. [1 M]

Answer :

Importance sampling allows us to correct the distribution of the returns from the behavior policy to match the target policy, thereby enabling us to estimate the value of s_1 under π using episodes generated by b.

(a-2) Compute the importance sampling ratio $\rho_{0:T-1}$ for each episode, where T is the episode's length. [2 M]

Answer :

The importance sampling ratio $\rho_{0:T-1}$ is the product of the ratios of the target policy to the behavior policy for each action taken in the episode, from the first step to the last step before the terminal state.

Episode 1: s_1, a_1, R = 1, s_2, a_1, R = 2, s_3 (terminal)
We compute $\rho_{0:T-1}$ by multiplying the ratios $\frac{\pi(a_t|s_t)}{b(a_t|s_t)}$ for each step:
- For s1 with action a_1: $\frac{\pi(a_1|s_1)}{b(a_1|s_1)}=\frac{0.7}{0.4}$ = 1.75
- For s2 with action a1: $\frac{\pi(a_1|s_2)}{b(a_1|s_2)}=\frac{0.9}{0.5}$ = 1.8

Thus, the importance sampling ratio for Episode 1 is $\rho_{0:T-1} = 1.75 \times 1.8 = 3.15$.

Episode 2: s1, a2, R = 0, s3 (terminal)
For this episode, the importance sampling ratio is computed as:
- For s_1 with action a_2: $\frac{\pi(a_2|s_1)}{b(a_2|s_1)}=\frac{0.3}{0.6}$ = 0.5

Thus, the importance sampling ratio for Episode 2 is $\rho_{0:T-1} = 0.5$.

(a-3) Using ordinary importance sampling, estimate $v_\pi(s_1)$ based on the two episodes.[2M]

Answer :

Use ordinary importance sampling to estimate $v_\pi(s_1)$, the value of state s_1 under the target policy π:
The formula for **ordinary importance sampling is**:

$$v_\pi(s_1) = \frac{\sum_{i=1}^{N} \rho_{0:T-1}^{(i)} \times G^{(i)}}{\sum_{i=1}^{N} \rho_{0:T-1}^{(i)}}$$

Episode 1: $G_1 = 1 + 2 = 3$
Episode 2: $G_2 = 0$ (because R = 0)

Using the importance sampling ratios:
$v_\pi(s_1) = \frac{(3.15 \times 3 + 0.5 \times 0)}{(3.15 + 0.5)} = \frac{9.45}{3.65} \approx 2.59$

Thus, the estimated value of sl under the target policy π is approximately $v_\pi(s_1)$ = 2.589.

(b) Consider a reinforcement learning agent using the TD(0) algorithm. The current value of state s, V(s), is 10. After taking action, the agent receives an immediate reward R=4 and transitions to state s′ with a value V(s′)=8. The learning rate α is 0.5, and the discount factor γ is 0.9. Calculate the updated value of V(s) using the TD(0) update rule. [2 M]

Answer :

The TD(0) value update formula is:

V(S)= $V(s)+\alpha[R+\gamma V(s')-V(s)]$

V(s)= 10+0.5[4+0.9*8-10] = 10+0.6= 10.6

Q4) [2 +2+2 = 6 Marks]

(a) Given the following Q-values and episodes from a grid-world environment (with actions at each grid are S, E, N, W) , apply the Q-learning update rule to calculate the updated Q-value for **Q ((1,2), E))** after Episode 2, with a learning rate α=0.5 and a discount factor γ=0.9. Use the Q-learning update equation:

$$Q(s_t, a_t) = (1 - \alpha)Q(s_t, a_t) + \alpha(r_t + \gamma \max_a Q(s_{t+1}, a))$$

Estimated Q-values of some of the grids after considering Episode -1 are as below:

- Q((1,3),S)=0
- Q((1,2),E)=0
- Q((2,2),E)=40
- Q((3,2),N)=50
- Q((3,2),S)=30

The Episode 2 is (assume the standard notation to interpret the episode) :

- (1,2), E, 0, (2,2)

(a-1) What is the updated Q-value for Q((1,2),E)? [2 M]

Answer :

1. **State (1,2), Action E**:
   - The agent takes action E from state (1,2) and transitions to state (2,2).
   - The reward r_t=0.
2. **Max Q-value for the next state (2,2)**:
   - The agent moves to (2,2), and from the Q-values provided, we have:
   - Q((2,2),E)=40.
   - So, $\max_a Q(s_{t+1},a)$=40
3. **Apply the Q-learning update**: Using the update rule:

$$Q((1,2),E)=(1-0.5)\times Q((1,2),E)+0.5\times(0+0.9\times 40)$$

Q((1,2),E)=0.5×0+0.5×(0+36)

Q((1,2),E)=0.5×36=18

(a-2) What happens if the value of α=0 and α=1? Explain using the above use case scenario [2M]

Answer :

alpha = 0 - no change in q values - no learning

alpha= 1- no exploitation - too much of episodic memory

(b) In TD learning, explain how the value update is performed using multiple steps of experience. How does n-step TD differ from traditional 1-step TD regarding bias and variance? [2 M]

Answer :

In n-step Temporal Difference (TD) learning, the value update is based on the rewards accumulated over the next n steps, rather than just the immediate next reward as in 1-step TD. The update equation incorporates the sum of rewards over n steps and the value of the state after n steps.

Bias and Variance:

1. n-step TD reduces bias compared to 1-step TD by considering more rewards, providing a more accurate estimate.
2. n-step TD has higher variance than 1-step TD because it uses more steps, leading to potentially more fluctuating updates. Thus, n-step TD balances bias and variance depending on the chosen n.

Q5) [1.5+1.5+1+2+1 = 7 Marks]

(a) **Usecase** : An RL agent learns to buy and sell stocks to maximize profit. The state space could involve financial indicators like stock prices, moving averages, and market sentiment.

(a-1) List the possible challenges when using look up table methods for the above mentioned scenario. [1.5 M]

Answer :

Tabular methods store exact value estimates (like Q-values or state-values) for every possible state-action pair in a lookup table, making them precise but impractical for large or continuous state spaces. Tabular methods struggle with scalability in high-dimensional environments, demanding extensive memory and computation resources. Tabular methods also struggle with partial observability—if the agent can't distinguish states precisely, the table becomes unreliable.

(a-2) How will a parameterized value function help you overcome the challenges with look-up tables? [1.5 M]

Answer :

Function approximation replaces the table with a parameterized function (e.g., a neural network) that generalizes across states, sacrificing exactness for scalability. Function approximation handles complexity by learning patterns from data. Function approximation addresses scalability by replacing the table with a model that predicts values.

(b)

(b-1) Differentiate Value-based and policy-based methods in terms of handling different types of actions, spaces and policies. [1 M]

Answer :

Value-based	Policy Based
Efficiency in Discrete Spaces	Continuous Action Spaces
Implicit Policy Representation	Stochastic Policies Explicit Policy Representation

(b-2) You are planning to use the Advantage Actor critic method for creating personalized treatment plans for patients in a healthcare scenario. Explain the roles of actor and critic in the actor-critic algorithm. How would actor-critic algorithm benefit over REINFORCE? [2M]

Answer :

Actor - suggesting treatment course
Critic - evaluating the diagnosis

REINFORCE suffers from high variance issues

(b-3) Which algorithm best suits the above scenario - actor-critic or Q learning? Justify your answer. [1 M]

Answer :

Actor- Critic algorithm
Efficiency: Handles continuous action spaces better.
Stability: Less prone to oscillations.
Flexibility: Can learn off-policy.

Insights: Learns both policy and value function.

Q6) [1+1+1+2 = 5 Marks]

(a) Explain the significance of supervised learning in AlphaGo [1M]
(b) How are the parameters of RL Policy network in AlphaGo initialised [1M] and what is the significance of its initialization [1M]
(c) Provide two real life examples which are better modelled as imitation learning. Explain why .[2 M]

Answer :

(a) Supervised Network (SLN) is used to learn the human moves of professional players which learns to imitate the human players. Instead of randomly exploring all exhaustive moves, SLN enables to speed up the learning game patterns. Using the learnt pattern policy network generalizes the play strategies and latter learns to optimize
(b) RL policy networks are initialized using the weights learnt from the supervised learning network. This enables heuristic based learning of better moves by the policy network. Here heuristics is derived from the human expert's historical moves

(c) Example 1: Autonomous Driving

Why Imitation Learning?

- Problem: Teaching a self-driving car to navigate roads, follow traffic rules, and handle diverse real-world situations (e.g., lane changes, yielding, stop signs).
- Imitation Learning Solution: Train the car by observing human expert drivers (e.g., from dashcam footage or driving logs).
  Reasoning: Human drivers already demonstrate safe and effective driving behavior, making it ideal to mimic expert policies.

Example 2: Robotic Surgery Assistance

Why Imitation Learning?

- Problem: Training a surgical robot to assist or perform tasks like suturing, cutting, or holding instruments during an operation.
- Imitation Learning Solution: The robot watches expert surgeons and learns how to replicate their actions precisely.
  Reasoning: Learning from human demonstrations is faster and more practical than hand-coding or reward-based learning.

Additional note to TA on Marking Scheme:

(a) Human Move & Speed of the learning
(b) Supervised learning net weights is the key
(c) Any 2 example where the agent can mimic the expert. 1 mark for each example.