

Time : 2-30 Hours]

**Instructions:** All questions are compulsory. Use of non-programmable scientific calculator is allowed.

- Q.1** (a) What is reinforcement learning? Explain the elements of reinforcement learning. (07)  
 (b) Explain bandit algorithm and its role in decision making. (07)

OR

- (a) Explain Markov Decision Process (MDP) with an example. (07)  
 (b) How reinforcement learning differs from other machine learning approaches. (07)

- Q.2** (a) Discuss how the bandit framework can be extended to handle contextual bandits and non-stationary environments. Provide examples of real-world applications where such extensions are essential. (07)  
 (b) Consider an MDP with two states,  $s_1$  and  $s_2$ , and two actions,  $a_1$  and  $a_2$ . The transition probabilities and rewards are as follows: (07)

Transition $P(s' s_1, a)$	Probability	Reward	Transition $P(s' s_2, a)$	Probability	Reward
$P(s_1 s_1, a_1) = 0.5$		7	$P(s_1 s_2, a_1) = 0.4$		-2
$P(s_2 s_1, a_1) = 0.5$		12	$P(s_2 s_2, a_1) = 0.6$		3
$P(s_1 s_1, a_2) = 0.7$		0	$P(s_1 s_2, a_2) = 0.6$		1
$P(s_2 s_1, a_2) = 0.3$		0	$P(s_2 s_2, a_2) = 0.4$		-2

Assume a discount factor  $\gamma = 0.9$ . Given the initial estimates:  $V(s_1) = 1$  and  $V(s_2) = 4$ , compute the updated value function numerically using one iteration of the Bellman update for the fixed policy  $\pi$  that always takes action  $a_1$  in both states. Solve the Bellman equations algebraically to find the exact values of both states.

OR

- (a) What is the Bellman Equation? How is it helpful in reinforcement learning? (07)  
 (b) Discuss the Generalized Policy Iteration (GPI) framework, outlining its key components and the overall process. (07)

- Q.3** (a) What are the main challenges in directly applying Monte Carlo methods for estimating action value function  $Q(s, a)$ ? How can these challenges be addressed? (07)  
 (b) What is importance sampling? Provide a detailed description of the two importance sampling methods used in offline learning. (07)

OR

- (a) Compare and contrast between Dynamic programming and Monte Carlo methods. (07)  
 (b) Explain the need for function approximation in Reinforcement Learning. Compare and contrast it with tabular methods. Also, provide examples of where function approximation is crucial in real-world RL problems. (07)

- Q.4** (a) Explain what is meant by *state aggregation* and how it simplifies the task of approximating value functions in large state spaces. (07)

P.T.O

E1328 2

- (b) You are given an environment with 1 state,  $x$ , and 2 actions,  $b$  and  $c$ .  $T$  is the terminal state. Your TD algorithm generates following episode using the policy  $\pi$ . (07)

Timestep	Reward	State	Action
0		$x$	$b$
1	14	$x$	$c$
2	10	$x$	$b$
3	14	$T$	

- The policy  $\pi$  is given by:  $\pi(b|x) = 0.8$ ,  $\pi(c|x) = 0.2$ .
- The current values of  $q$  are:  $q(x, b) = 1$  and  $q(x, c) = 2$ .
- Discount factor,  $\gamma = 0.5$  and step size  $\alpha = 0.1$ .

Show the values of  $q(x, b)$  and  $q(x, c)$  after their first update using 1-step SARSA, 2-step SARSA, and 2-step Expected SARSA. Also, compute first update for  $q(x, b)$  using Q learning.

OR

- (a) Describe the *Tile Coding* technique used for function approximation. Explain how it handles continuous state spaces. (07)
- (b) Explain the concept of *Experience Replay* in reinforcement learning. Also, discuss the advantages and disadvantages of experience replay. (07)

Q.5 Attempt any SEVEN out of TWELVE: (14)

- (1) Differentiate between episodic tasks and continuing tasks.
- (2) If Arm A has an average reward of 3.5 after 10 pulls, and Arm B has an average reward of 5.5 after 5 pulls, which arm would you select using the upper confidence bound (UCB) approach, assuming both have been pulled for at least one round? Take  $c = 2$ .
- (3) State Markov property.
- (4) Suppose  $\gamma = 0.9$  and the reward sequence is  $R_1 = 2$  followed by an infinite sequence of 7s. What are the values for  $G_1$  and  $G_0$ ?
- (5) Briefly describe what bootstrapping is in the context of Reinforcement Learning.
- (6) What is a policy, and how does it differ from transition probability?
- (7) In the context of reinforcement learning, explain (i) on-policy, (ii) off-policy.
- (8) What are the advantages of using TD learning over dynamic programming?
- (9) When using TD learning, why is it better to learn action values (Q-values) rather than state values (V-values)?
- (10) Why does double Q learning use double learning?
- (11) Why do we need a policy-based method?
- (12) What does a high entropy coefficient encourage in a learned policy?

\*\*\*\*

\*\*\*\*