Mahindra University Hyderabad
École Centrale School of Engineering
Mock Paper for End Semester Examination

Program: B. Tech. Branch: AI/CSE/ECM/CAM Year: IV Semester: I
Subject: Reinforcement Learning and Autonomous Systems (CS 4122)

Time Duration: 3 hours **Max. Marks: 100**

Instructions:

1) The submitted answer sheet should only contain your final answer.
2) Use separate sheets for rough work. Don't submit rough work.
3) You must show your solution procedure in the submitted answer sheet.

ALL FIGURES AND TABLES ARE IN PAGE 4.

Q1: Conceptual Questions (20 marks, Difficulty: 7/10, exam question is easier.)

(a) *Before starting this question, a suggestion: You may want to remove the stapler from the question paper and have Figures 1-a to 1-c in front of you.*
Consider the probabilistic graphical model (PGM) shown in Figure 1-a. This PGM does NOT follow Markovian property because random variable x_t depends on both x_{t-1} and x_{t-2} (it would have followed Markovian property if x_t depended only on x_{t-1}). Figure 1-b shows convert PGM in Figure 1-a to an equivalent PGM that follows Markovian property. The trick is to define a new random variable z_t that contains the necessary history of random variables such that z_t follows Markovian property. Now answer the following questions:

- **i.** Is the PGM in Figure 1-b of mulit-armed bandits, contextual bandits, or Markov decision process? Justify your answer using no more than three sentences. **(2 marks)**
- **ii.** You are given a PGM in Figure 1-c that does NOT follow the Markovian property. Using the conversion of PGM in Figure 1-a to 1-b as an example, draw an equivalent PGM for Figure 1-c that follows the Markovian property. **(5 marks)**

(b) Write the set of formulas used to update the parameters of linear conextual bandits. You answer must comprise of the formula and the definition of the notations. NOTE: DON'T write the entire psuedocode for ε-greedy and UCB algorithm! Also, DON'T write the formula to choose the actions in ε-greedy and UCB algorithm. **(5 marks)**

(c) A Markov decision process has the following trajectory τ, **(8 marks)**

$$\tau = \{(x_0, a_0, r_0), (x_1, a_1, r_1), (\boldsymbol{x_2}, a_2, \boldsymbol{r_2})\}$$

The following are known:

- x_0, a_0, r_0, x_1, a_1, r_1, a_2.
- Transition probability distribution $f_T(x'|x, a)$.
- Reward probability distribution $f_R(r|x, a)$.

- Policy $\pi(a|x)$.

$\boldsymbol{x_2}$ and $\boldsymbol{r_2}$ are NOT known. We want to find the most probable x_2. For this, it is important to know the probability distribution of x_2. Find the probability distribution of x_2 **given** $\boldsymbol{x_0}, \boldsymbol{a_0}, \boldsymbol{r_0}, \boldsymbol{x_1}, \boldsymbol{a_1}, \boldsymbol{r_1}, \boldsymbol{a_2}$. This probability distribution should be in terms of the known quantities. HINT: Use law of total conditional expectation, and something similar to Baye's rule.

Q2: Advanced Reinforcement Learning (20 marks, Difficulty: 3/10 but lengthy)

This is going to be a direct theory question on DDPG or Inverse RL. So, **learn both. Don't end up writing everything in the notes**. Write to the point.

Q3: Reinforcement Learning (20 marks, Difficulty: 5/10, exam question is of similar difficulty.)

Consider an episodic reinforcement learning environment with two states x and y, and three actions L, R, and M. Modified SARSA with learning rate of 0.5 and discount factor of 1 is used to train a policy for this environment. Modified SARSA estimates the Q-function as follows:

$$q(x_t, a_t) \approx r_t + \beta r_{t+1} + \beta^2 q(x_{t+2}, a_{t+2})$$

Q-value of all state-action pairs are initialized to ZERO. Table 1 shows a trajectory of episode 1. Answer the following questions. *Suggestion: You may want to remove the stapler from the question paper and have Table 1 in front of you.*

(a) Estimate the Q-values of each state-action pair at the end of episode 1. Show necessary steps of calculation. **(6 marks)**

(b) In which all time slots of episode 1 did the policy explore? **(4 marks)**

(c) Use your answer for part (b) to get an estimate of the exploration probability ε using the sample average method. **(4 marks)**

(d) Consider that for the given reinforcement learning environment, the state transitions and rewards are deterministic. Come up with an algorithm to find the optimal policy in finite number of episodes. **(6 marks)**

Q4: Markov Decision Process (20 marks, Difficulty: 6/10, exam question is slightly difficult provided you solved programming assignments 1 to 4. Otherwise, it is very difficult.)

You are working in a company to design an autonomous car. The $\boldsymbol{x}$ denote the state associated with the autonomous car. x can include the position, velocity, steering angle, lane information etc. Let $\boldsymbol{a} = (\boldsymbol{a_1}, \boldsymbol{a_2})$ denote the action associated with the autonomous car where $\boldsymbol{a_1}$ is the acceleration $\boldsymbol{a_2}$ is the steering angle. Let the reward for state-action pair (x, a_1, a_2) be $\boldsymbol{r}(\boldsymbol{x}, \boldsymbol{a_1}, \boldsymbol{a_2})$. The state transition probability is $\boldsymbol{f_T}(\boldsymbol{x}'|\boldsymbol{x}, \boldsymbol{a_1}, \boldsymbol{a_2})$.

The company consists of two teams: *Team 1* and *Team 2*. Team 1 is in-charge of designing the policy for action a_1 (acceleration). And, Team 2 is in-charge of designing the policy for action a_2 (steering angle). You are part of team 2. Now, team 1 has already finished designing the policy for a_1. The policy is $\boldsymbol{\pi_1(a_1|x)}$. **Team 2 is aware of this policy and it can't be changed whether or not this policy will lead to a sub-optimal performance of the overall system**. As part of team 2, your task in this question is to design an optimal policy for a_2. Answer the following questions:

(a) What are the equations governing state transition given that a_1 is decided by π_1? **(6 marks)**

(b) Write down the Bellman optimality equation to find the optimal policy for a_2 given that action a_1 is decided by π_1. **(8 marks)**

(c) Assume that team 2 has finished designing its optimal policy. Team 1 then assumes that team 2's policy is fixed and then improves it's policy. Then, team 2 again optimizes it's policy assuming team 1's policy is fixed and this cycle continues. Will this iterative process converge to an optimal policy for the entire system? Justify your answer. **(6 marks)**

Q5: Deep Reinforcement Learning (20 marks, Difficulty: 10/10, exam question is easier provided you understand the psuedocodes of contextual bandits and DQN.)

(a) Consider policy gradient and actor critic RL where $\nabla_\theta J(\theta)$ is the gradient of the expected discounted reward, $J(\theta)$, with respect to policy network parameter, θ. Give an argument to justify that the the variance of the estimate of $\nabla_\theta J(\theta)$ is higher for policy gradient RL compared to actor crictic RL. **(5 marks)**
HINT: How does causality helps in reducing the variance of the estimate of $\nabla_\theta J(\theta)$.

(b) While training Deep Q-Networks (DQN) it is usually assumed that the transition probability distribution and reward probability distribution are NOT known. But in some setups, transition probability distribution is known. This question is targeted towards such setups. More specifically for DQN architecture 1, the target corresponding to a sample $\langle x, a, r, x' \rangle$ is,

$$y = r + \beta \max_{a'} \hat{q}(x', a'; \phi_T)$$

Recall that the target y is an estimate of the Q-value for state-action pair (x, a). This estimate can be improved when the transition probability distribution is known.

Find an approach to generate target which is a better estimate of the Q-value when transition probability distribution is known. This approach should also account for the fact that the state space can be huge and hence summing over all the states in the state space is not possible. Then use this approach to write a psuedocode to train DQN architecture type 1. *Note:* **(15 marks)**

- The pseudocode along with the necessary explanation must be limited to 2 pages of your answer script. DON'T use Double Q-Learning.

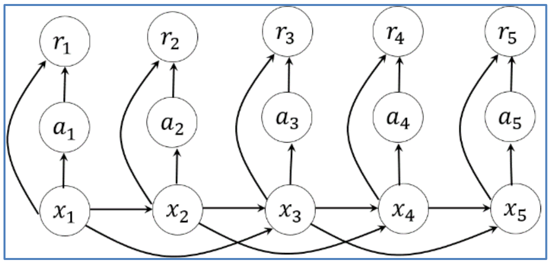

Figure 1-a

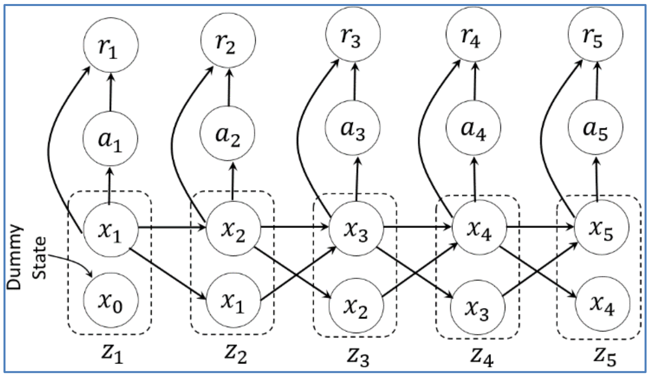

Figure 1-b

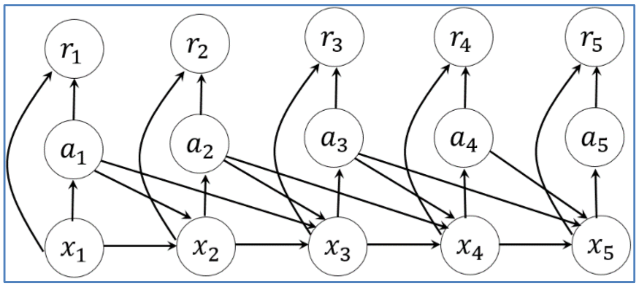

Figure 1-c

Table 1

Time slot	0	1	2	3	4	5	6
State	y	x	y	x	y	x	y
Action	R	L	M	L	L	L	M
Reward	-1	0	2	-1	0	-1	2