

**Mahindra University Hyderabad**  
**École Centrale School of Engineering**  
**Minor-II**

**Subject: Reinforcement Learning and Autonomous Systems (CS 4122)**

**Date: 25/10/2024**  
**Time Duration: 1 hour 30 minutes**

**Start Time: 10:00 am**  
**Max. Marks: 30**

**Instructions:**

- 1) The submitted answer sheet should only contain your final answer and the **solution procedure**. Use separate sheets for rough work.
- 2) The paper has a sum total of **40 marks**. However, you can score a maximum of **30 marks**. Any **bonus marks** you score will be included in the next exam.

**Q1: A Variant of SARSA (10 marks, Difficulty: EASY)**

The following shows how Q-function is estimated in SARSA:

$$\begin{aligned} q(x_t, a_t) &\approx r_t + \beta r_{t+1} + \beta^2 r_{t+2} + \beta^3 r_{t+3} + \dots \\ &\approx r_t + \beta(r_{t+1} + \beta r_{t+2} + \beta^2 r_{t+3} + \dots) \\ &\approx r_t + \beta q(x_{t+1}, a_{t+1}) \end{aligned}$$

An alternate approach to estimate the Q-function is as follows:

$$\begin{aligned} q(x_t, a_t) &\approx r_t + \beta r_{t+1} + \beta^2 r_{t+2} + \beta^3 r_{t+3} + \dots \\ &\approx r_t + \beta r_{t+1} + \beta^2(r_{t+2} + \beta r_{t+3} + \dots) \\ &\approx r_t + \beta r_{t+1} + \beta^2 q(x_{t+2}, a_{t+2}) \end{aligned}$$

Write a neat psuedocode for SARSA using this alternate approach of estimating the Q-function. **I only need the psuedocode. You will loose points if you give any other explanation.**

**Q2: Bandit Setup (5 marks + 10 marks)**

Answer the following questions:

- (a) **(Difficulty: EASY):** There are two ways in which training a neural network for policy gradient algorithm is different from training a neural network for classification problems in supervised learning. What are those two ways? **Your answer must not exceed ¼ of a page.**
- (b) **(Difficulty: HARD):** There is a algorithm for bandits called “explore then commit”, abbreviated as ETC. ETC works as follows: The exploration phase starts first. In this phase, each of the actions are selected exactly M times. After the exploration phase ends, the commit phase begins. In commit phase, the action with highest sample average reward is choosen.

We use ETC algorithm for a MAB setup where it is known that the reward of each action follow a Gaussian distribution with mean between 1 to 5 and variance between 0.25 to 4. The exact mean and variance of each action is NOT known. The hyperparameter  $M$  of ETC should be chosen such that the action value of each action is known to a certain degree of accuracy before the commit phase begins. What should be the value of  $M$  such that the percentage error between the true mean reward and sample average reward for each action is less than 5% with a probability of 0.95? **HINT:** Recall the formula for mean and variance of sample average (of iid samples). You should have studied this in your Probability course.

### Q3: MDP for Cloud Computing (15 marks, Difficulty: MODERATE if you did programming assignment 2)

In cloud computing, there are two kind of computing instances (virtual machines):

1. *On-demand instances:* On-demand instances has a fixed price but is guaranteed, i.e. if an user wants it, the user can buy it for sure buy paying a price of  $\alpha$ .
2. *Spot instances:* To buy a spot instance, an user has to take part in an auction. Spot instances are generally cheaper than on-demand instances but winning the auction is not guaranteed. If the user bids  $\theta$  in the auction, where  $\theta \in \{\delta, 2\delta, \dots, N\delta\}$ , the probability of winning the auction is  $\rho_\theta$  (obviously  $\rho_\theta$  increases with  $\theta$ ). If the user wins the auction, it has to pay  $\theta$  and it will get to use a spot instance. Remember,  $\rho_\theta < 1$  for all  $\theta$ , i.e. there is always a risk of not winning the auction.

We consider that time is divided into time slots. In the beginning of every time slot, an user either gets a demand,  $d = 1$ , or not get a demand,  $d = 0$ . Let  $d$  and  $d'$  be the demand in the current and next time slot. Then probability that the current demand being  $d$  and the next demand being  $d'$  is  $p_{d,d'}$ . In order to process the demand in the current time slot, the user has to decide whether it wants to buy an on-demand or a spot instance at every time slot. Also, if it decides to buy a spot instance, then how much to bid. Both types of instances if purchased will last for one time slot only. The objective of the user is to minimize the discounted cost of buying instances but with a hard constraint: The user is allowed to not process a demand in a time slot but the number of demands not processed in the last  $\tau$  time slots can't be greater than  $\eta$ , where  $\eta < \tau$ . Answer the following questions:

- (a) What is the state and state space for this problem? (3 marks)
- (b) What is the action and action space for this problem? (3 marks)
- (c) Define the reward for all state-action pair for this problem? (3 marks)
- (d) What is the Bellman optimality equation for this problem? (6 marks)