IOC5269 Spring 2021: Reinforcement Learning **(Due: 2021/04/07, 21:00pm)**

Homework 1–Part II: Policy Gradient and Model-Free Prediction

Submission Guidelines: Please compress all your write-ups (photos/scanned copies are acceptable; please make sure that the electronic files are of good quality and reader-friendly) and source code into one .zip file and submit the compressed file via E3.

Problem 1 (Baseline for Variance Reduction) (10+10+15=35 points)

Consider an example similar to that in the slides of Lecture 8 for explaining the baseline. Suppose there are only 1 non-terminal starting state (denoted by s) and 3 actions (denoted by a, b, c) in the MDP of interest. After any one of the action is applied at the starting state s, the MDP would evolve from s to the terminal state, with probability 1. Moreover, consider the following setting:

- The rewards are deterministic, and the reward function is defined as $r(s, a) = 100$, $r(s, b) = 98$, and $r(s, c) = 95$. Moreover, there is no terminal reward.
- We consider a softmax policy with parameters $\theta_a, \theta_b, \theta_c$ such that $\pi_\theta(\cdot|s) = \exp(\theta_\cdot)/(\exp(\theta_a) + \exp(\theta_b) + \exp(\theta_c))$. Moreover, currently the parameters are $\theta_a = 0$, $\theta_b = \ln 5$, $\theta_c = \ln 4$.
- We would like to combine PG with SGD. At each policy update, we would construct an unbiased estimate $\widehat{\nabla} V$ of the true policy gradient $\nabla_\theta V^{\pi_\theta}$ by sampling one trajectory (Note: $\widehat{\nabla} V$ is a random vector. In this example, each trajectory has only one time step, and $s_0 = s$, a_0 is either a, b, or c, and s_1 is the terminal state).

(a) What are the mean vector of $\widehat{\nabla} V$ (denoted by $\mathbb{E}[\widehat{\nabla} V]$) and the covariance matrix of $\widehat{\nabla} V$ (i.e., $\mathbb{E}[(\widehat{\nabla} V - \mathbb{E}[\widehat{\nabla} V])(\widehat{\nabla} V - \mathbb{E}[\widehat{\nabla} V])^\intercal]$)?

(b) Suppose we leverage the value function $V^{\pi_\theta}(s)$ as the baseline and denote by $\tilde{\nabla} V$ the corresponding estimated policy gradient. Then, what are the mean vector and the covariance matrix of $\tilde{\nabla} V$? (Note: $\tilde{\nabla} V$ is also a random vector)

(c) Let $B(s)$ denote a baseline function and ∇V_B be the corresponding estimated policy gradient (∇V_B is again a random vector). Suppose we say that a baseline function $B(s)$ is *optimal* if it attains the minimum trace of the corresponding covariance matrix of ∇V_B among all possible state-dependent baselines. Please try to find one such optimal $B(s)$.

Problem 2 (Policy Gradient) (10+10=20 points)

(a) Show the following useful property discussed in Lecture 6: for any function $f : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$,

$$\mathbb{E}_{\tau \sim P_\mu^{\pi_\theta}}\Big[\sum_{t=0}^{\infty} \gamma^t f(s_t, a_t)\Big] = \frac{1}{1-\gamma} \mathbb{E}_{s \sim d_\mu^{\pi_\theta}} \mathbb{E}_{a \sim \pi_\theta(\cdot|s)} \big[f(s, a)\big] \tag{1}$$

(Hint: It might be slightly easier to go from the RHS to LHS. Specifically, you may first expand the RHS of (1) into a sum of $f(s, a)$ over s and a and then apply the definition of $d_\mu^{\pi_\theta}$, which involves a sum of probability over t. Next, try to reorganize the triple summation into the form of the LHS of (1))

(b) Show that for episodic environments, the policy gradient can be expressed using the advantage function A^{π_θ} as follows:

$$\nabla_\theta V^{\pi_\theta}(\mu) = \mathbb{E}_{\tau \sim P_\mu^{\pi_\theta}}\Big[\sum_{t=0}^{T-1} \gamma^t A^{\pi_\theta}(s_t, a_t) \nabla_\theta \log \pi_\theta(a_t|s_t)\Big] \tag{2}$$

Problem 3 (Implement Monte-Carlo and Temporal Difference Policy Evaluation) (50 points)

In this problem, we will apply both **first-visit MC** and **TD(0)** to reproduce the plots of the value function of the Blackjack problem under the fixed policy that one will stick only if the sum of cards is greater than or equal to 20. The Blackjack environment has been included in the OpenAI Gym: https://gym.openai.com/envs/Blackjack-v0/. Please read through **mc_td_policy_evaluation.py** and then implement the two functions **mc_policy_evaluation** and **td0_policy_evaluation** (Note: please set $\gamma = 1.0$).