(Fall 2024) 515512 Probability **(Early Bird: 2024/11/29, 9pm; Normal: 2024/11/30, 9pm)**

Homework 3: Joint Distributions, MGFs, and Concentration Inequalities

Submission Guidelines: You will submit two files via E3: (1) Please compress all your technical report and write-ups (photos/scanned copies are acceptable; please make sure that the electronic files are of good quality and reader-friendly) into one .pdf file (2) Please also submit your Jupyter Notebook files.

Problem 1 (Joint Distributions of Discrete Random Variables) (6+6+6=18 points)

Shinemood is a popular waffle store located on campus in NYCU. Recently, Shinemood started selling a special type of Uji Matcha waffles. Due to the additional efforts needed in making Uji Matcha, on each day, Shinemood only sells a limited number of Uji Matcha waffles, which is determined by a modified Poisson random variable X as follows:

- On each day, construct $Z \sim \text{Bernoulli}(p)$ to be a Bernoulli random variable. Moreover, generate a Poisson random variable $Y \sim \text{Poisson}(\lambda, 1)$, where λ is the rate parameter. Suppose Y and Z are independent.
- Define X as follows: (i) If $Z = 0$, then $X = 0$. (ii) If $Z = 1$, then $X = Y$. (Note: Such X is often called a *Zero-Inflated Poisson* random variable)

(a) Write down the PMF of X (Note: When you are specifying a formula for the PMF, make sure to specify the range over which the formula holds).

(b) Define another random variable $\widetilde{X} := (1 - I) \cdot Y$, where $I \sim \text{Bernoulli}(1 - p)$ is independent of Y. Please clearly explain why $\widetilde{X}$ and X have the same PMF.

(c) Show the following useful property: For any two *independent* discrete random variables U, V (for which $E[|U|] < \infty$, $E[|V|] < \infty$, and $E[|UV|] < \infty$), we always have $E[UV] = E[U] \cdot E[V]$.

Problem 2 (Moment Generating Functions) (6+8=14 points)

(a) Suppose X and Y are i.i.d. Poisson random variables with rate $\lambda = 2$ and observation window $T = 1$. Use MGFs to determine whether $3X + 4Y$ is also a Poisson random variable?

(b) Let Y be a discrete random variable with PMF

$$p_Y(k) = \begin{cases} \frac{6}{\pi^2 k^2}, & \text{if } k \in \mathbb{N} \\ 0, & \text{otherwise} \end{cases}$$

Show that the MGF of Y (denoted by $M_Y(t)$) does NOT exist, i.e., there exists no interval of the form $(-\delta, \delta)$ (with $\delta > 0$) such that $M_Y(t)$ exists. (Hint: Show that $M_Y(t)$ is not finite on $t \in (0, \infty)$)

Problem 3 (Joint Distributions of Continuous Random Variables) (8+6=14 points)

Let X, Y be two random variables with the joint CDF

$$F_{XY}(t, u) = \begin{cases} 1 - \exp(-t) - \exp(-u) + \exp(-(t + u + \theta tu)), & \text{if } t > 0, u > 0 \\ 0, & \text{otherwise} \end{cases}$$

where $\theta \in [0, 1]$. Please try to derive the following properties of X and Y.

(a) Find the marginal CDF of X, Y. For which values of θ (if any) are X, Y independent?

(b) Find the joint PDF of X, Y.

Problem 4 (Sum of Independent Random Variables and Chernoff Bounds) (8+8+8=24 points)

In this problem, from a probabilistic perspective, we study the fine-tuning method behind ChatGPT – *Reinforcement Learning from Human Feedback* (RLHF). Specifically, under RLHF, ChatGPT is fine-tuned by repeatedly applying the following *pairwise text comparison by human labelers* :

- Each piece of text, denoted by $\mathcal{T}$ has some underlying true score $R(\mathcal{T})$, which is a scalar for specifying the quality of the text (the higher the score, the better the quality).
- Define the Sigmoid function as $\sigma(x) := 1/(1+\exp(-x))$. It is easy to verify that (i) $\sigma(x) \in (0,1)$ for all $x \in \mathbb{R}$ and (ii) $\sigma(x) = 1 - \sigma(-x)$ for all $x \in \mathbb{R}$.
- At each trial, two pieces of texts $\mathcal{T}_a, \mathcal{T}_b$ (with true scores $R(\mathcal{T}_a)$ and $R(\mathcal{T}_b)$, respectively) are provided to a human labeler for comparison. Under the RLHF model, the human labeler would output one of the following two responses: (i) The labeler would vote for "$\mathcal{T}_a$ is better than $\mathcal{T}_b$", with probability $\sigma(R(\mathcal{T}_a) - R(\mathcal{T}_b))$. (ii) The labeler would vote for "$\mathcal{T}_b$ is better than $\mathcal{T}_a$", with probability $\sigma(R(\mathcal{T}_b) - R(\mathcal{T}_a))$.
- All the comparison trials by the human labelers are assumed to be independent.

(a) Suppose our CS department would like to train a new GPT model, called NYCU-GPT, by using RLHF. We hire n human labelers (n is odd) and provide the same two pieces of texts (denoted by $R(\mathcal{T}_a)$ and $R(\mathcal{T}_b)$) to these n labelers. Suppose $R(\mathcal{T}_a) > R(\mathcal{T}_b)$. Define an event $E := \{\mathcal{T}_a \text{ get more votes than } \mathcal{T}_b\}$. Then, by using Hoeffding's inequality, what could we say about $P(E)$? (Please provide a lower bound for $P(E)$).

(b) Based on the result in (a), suppose we would like to ensure that $P(E) \geq 1 - \delta$, where δ is a small positive constant. Then, by Chebyshev's inequality, how many human labelers do we need to achieve this? Similarly, suppose we use the Hoeffding's inequality for the analysis instead. How many human labelers do we need to ensure that $P(E) \geq 1 - \delta$?

(c) For further training, we proceed to prepare a collection of 100 pieces of texts, denoted by $\mathcal{T}_1, \cdots, \mathcal{T}_{100}$. Suppose we already know that the scores satisfy that $R(\mathcal{T}_1) > R(\mathcal{T}_2) > \cdots > R(\mathcal{T}_{100})$. Then, we ask a human labeler (whose name is Sam Altmann), to finish pairwise text comparison for all the pairs of these 100 pieces of texts. Define random variables $I_j := \mathbb{I}\{\text{Sam thinks } R(\mathcal{T}_j) \text{ is better than } R(\mathcal{T}_1)\}$ (where $\mathbb{I}\{A\}$ is the indicator function of an event A, i.e., $\mathbb{I}\{A\}$ would output 1 if the event occurs and output 0 otherwise.) Please use the Chernoff technique to find an upper bound of $P(I_2 + I_3 + \cdots + I_{100} > 90)$. (Hint: Note that we could NOT directly apply Hoeffding's inequality here since $\{I_j\}$'s are independent but NOT identically distributed. Despite this, we may still use the Chernoff technique to find an upper bound by leveraging independence)

Problem 5 (Programming: Monte-Carlo Method) (8+8=16 points)

In this problem, let us consider a very interesting and somewhat surprising approach to estimate the Euler's number $e = 2.71828\cdots$ via the Monte Carlo method. Specifically, given that e is a deterministic number, we need a way to connect e with some random variable as follows:

(a) Let $U_1, U_2, \cdots$ be a sequence of i.i.d. continuous uniform random variables between 0 and 1. Define the running sum $S_n = \sum_{i=1}^{n} X_i$. Moreover, define N to be the smallest integer such that $\sum_{i=1}^{N} U_i > 1$. Clearly, N is also a random variable. Then, we would like to show that $E[N] = e$. To prove this, we need to first write down the PMF of N through the following steps:

- Step 1: Show that $P(N = n) = P(S_n > 1 \text{ and } S_{n-1} < 1) = P(S_{n-1} < 1) - P(S_n < 1)$.
- Step 2: Show that $P(S_n < 1) = 1/n!$, for all $n \in \mathbb{N}$.
- Step 3: Combine Step 1 and Step 2 and find $P(N = n)$ as well as $E[N]$.

Please follow the above steps and find $E[N]$.

(b) Based on (a), please write down the detailed procedure that describes how you can apply the Monte Carlo method to estimate the value e. Moreover, please also write a short Python program to implement your

procedure. You may use the sample code hw3_problem4_mc.py as the starting point. What are your estimates of e under $10^1, 10^3, 10^5$, and 10^7 sample trials?

Problem 6 (Programming: Multi-Armed Bandits) (16+10=26 points)

As described in Lecture 17, multi-armed bandits (MAB) a classic formulation for capturing exploration and exploitation trade-off in online machine learning. Let's briefly describe the MAB setting as follows:

- Consider the stochastic K-armed bandit problem, where each arm i is characterized by its reward distribution $\mathcal{D}_i$ with mean θ_i.
- At each time $t = 1, \cdots, T$, the decision maker chooses an arm denoted by $\pi_t \in \{1, \cdots, K\}$ and observes the corresponding random reward X_t, which is independently drawn from the distribution $\mathcal{D}_{\pi_t}$.
- Let $N_i(t)$ and $S_i(t)$ be the total number of plays of arm i and the total reward collected from pulling arm i up to time t, respectively. We define $p_i(t) := S_i(t)/N_i(t)$ as the empirical mean reward up to t.
- Based on the multi-armed bandit convention, our objective is to minimize the *regret* defined as

$$\mathcal{R}(T) := T \cdot \max_{i \in \{1,\cdots,K\}} \theta_i - \mathbb{E}\Big[\sum_{t=1}^{T} X_t\Big],$$

  where the expectation is taken with respect to the randomness of the rewards and the employed strategy of the decision maker.

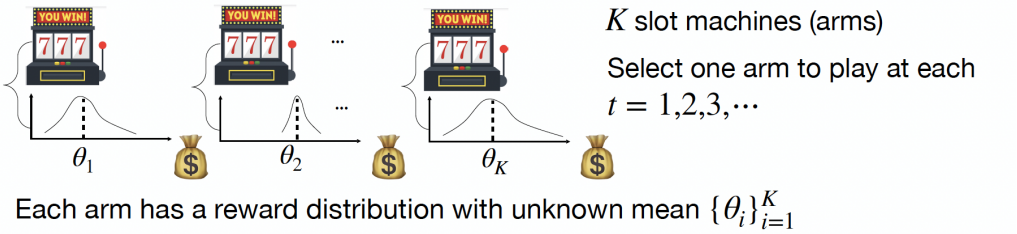

Figure 1: An illustration of the standard MAB problems.

To build an MAB environment as described above, we leverage the popular MAB package **SMPyBandits** (available at https://github.com/SMPyBandits/SMPyBandits and can be installed via pip).

In this homework problem, you will implement one classic and useful algorithm called *Epsilon-Greedy* algorithm and compare it with a naive *Empirical Means* algorithm (also known as the *Greedy* algorithm). Under the Epsilon-Greedy method, at each time t, the decision maker chooses an exploration parameter $\epsilon \in [0, 1]$ and enforces some exploration by:

- The decision maker first creates a Bernoulli random variable Z_t with success probability $1 - \epsilon$.
- (Exploitation) If $Z_t = 1$, then the decision maker chooses the arm with the largest empirical mean at time $t - 1$, i.e., $p_i(t - 1)$.
- (Exploration) If $Z_t = 0$, then the decision maker simply selects one of the K arms uniformly at random.

You will do this Python programming task on Jupyter Notebook as in HW2. Please take a look at the notebook "MAB.ipynb" and finish the tasks therein (normally you need no more than 20 lines of code in total).

(a) Please finish the remaining parts of "MAB.ipynb". What are the mean regrets of your Epsilon Greedy algorithm under the three MAB problem instances provided in "MAB.ipynb" and under $\epsilon = 0.01, 0.03, 0.1, 0.3$? Please also compare the performance of Epsilon Greedy algorithm with that of the EmpiricalMeans algorithm.

(b) Suppose we use a diminishing exploration rate in the Epsilon Greedy algorithm , i.e., $\epsilon(t) = t^{-\alpha}$ with $\alpha > 0$? Then, what are the mean regrets under $\alpha = 0.1, 0.5, 1.0$, and 2.0? Can you point out some interesting things from your experimental results?

Please briefly summarize your observation in a technical report (no more than 3 pages) and turn in your code and the report via E3.