EE1080/AI1110/EE2102 Probability and Random Processes

9th March, 2025

Please answer all the questions. The total marking is for 100. Rules for each question and split across the questions are described below. Please read the instructions carefully and do not wait until the deadline to work on the questions.

General Instructions for Submission

The format, naming convention for the files to be submitted is described below. *Submissions that do not follow the format below will not be considered for evaluation.*

- Create a separate Python file for each problem. For instance, the file for problem one should be named `1_Myna.py`, where **Myna** should be replaced with your first name. Follow a similar naming pattern for the remaining files.
- create a undertaking_Myna.pdf (replace Myna with your first name) with an undertaking that reads

  " I encoded this program myself, did not copy or rewrite the code of others, and did not give or send it to anyone else.

  Signature"

- Place all the Python files along with the undertaking in a `.zip` archive named `0_Myna_ee24btech00000.zip`, where:
  - **Myna** should be replaced with your first name,
  - **0** should be replaced with your course serial number used in attendence, and
  - **ee24btech00000** should be replaced with your roll number.

Additional points to note :

- Clearly indicate the **functions** or **code modules** that correspond to each part of the question for easier evaluation.
- Wherever possible, structure your code into **functions** or **modules**. For example, consider creating one function to **generate random samples** and another to **calculate the expected value** of a random variable.
- Add **comments** to your code to improve readability.
- If you use a formula directly, please **cite the source** (e.g., a Wikipedia page or textbook section) for reference.
- Use a **random seed** (e.g., `random.seed(42)`) in relevant problems to ensure your results are reproducible during evaluation.
- Follow proper **indentation** and adopt **good coding practices** to ensure your code is clean, organized, and easy to understand.

1 St. Petersburg paradox

A casino offers a game of chance for a single player in which a fair coin is tossed until the first time tails appears. If this happens at the k-th toss, the player wins 2^k dollars. (So, if the first toss is tail, the player wins 2 dollars, if the first toss if head, the second is tail, the player wins 4 dollars, but if the first 9 tosses are heads, and the 10-th is tail, the player wins $2^{10} = 1024$ dollars. Simulate m games (that is we play till the m-th tail). What is the average payout if $m = 100$, $m = 10000$ and if $m = 1000000$?

1.1 Requirements

The output of the program is the three average numbers with three digits rounded to three decimal places (after the decimal point). Please use this format, wherever you are asked to print any decimal values. The three numbers need to be separated by spaces. The aim of this task is to experience a random variable with infinite expected value. (We have seen this example in class when expectation definition is introduced).

1.2 Grading

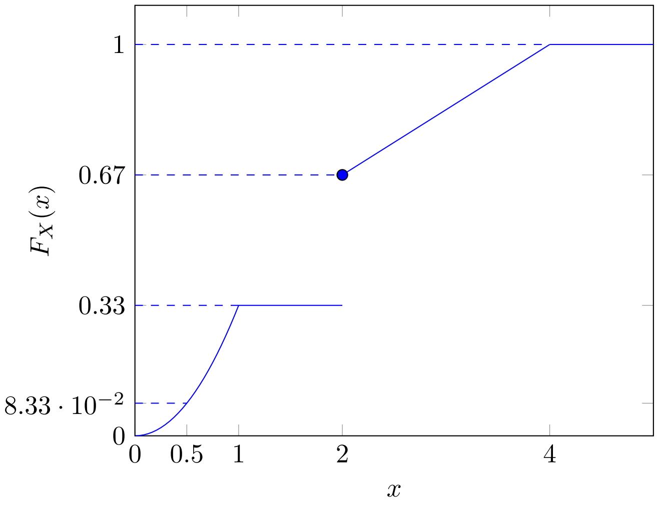
This question will carry 15 marks.

2 Using Uniform Samples to Generate Samples of other Random Variables

In this problem, you'll have to use N uniform $[0, 1]$ random variable samples to:

1. generate N samples of a Bernoulli(p) random variable.
2. generate N samples of an exponential(λ) samples.
3. generate N samples of a random variable X with cumulative distribution function (CDF), $F_X(x)$ given below:

$$F_X(x) = \begin{cases} x^2/3 & x \leq 1 \\ 1/3 & 1 < x < 2 \\ (x+2)/6 & 2 \leq x \leq 4 \end{cases}$$



2.1 Requirements

1. Input arguments to the this program:
   (a) first argument is mode it indicates, if it is 0, then your program should output Bernoulli samples, if 1, exponential samples and if 2, samples of X.
   (b) second argument is the samples file name. uniform_samples.txt test file is provided in the folder that you can use to test your code.
   (c) third argument is the parameter p for mode 0 and parameter λ for mode 1.
2. Example commands that you should test with:
   (a) `python 2_Myna.py 0 uniform_samples.csv 0.25`
   (b) `python 2_Myna.py 1 uniform_samples.csv 0.5`
   (c) `python 2_Myna.py 1 uniform_samples.csv 5.5`

    (d) python 2_Myna.py 2 uniform_samples.csv

3. Output of this program.

   (a) For all the modes, the input file needs to be read to get the uniform samples and the number of samples N need to be determined.

   (b) For all the modes you are expected to save the sample outcomes to a an output file with following naming convention:

      i. Mode 0: Bernoulli_{p_value}.csv. For example command shown in 2a, you should have the output file name as Bernoulli_p25.csv

      ii. Mode 1: Exponential_{λ_value}.csv. For example command shown in 2b, you should have the output file name as Exponential_p5.csv. For example command shown in 2c, you should have the output file name as Exponential_5p5.csv

      iii. Mode 2: CDFX_{p_value}.csv. For example command shown in 2d, you should have the output file name as CDFX.csv

   (c) For Mode 0, after the Bernoulli samples are generated from the uniform samples read from the file, you need to compute sample mean and print the value.

   (d) For Mode 1, after the exponential samples are generated from the uniform samples, you need to plot histogram with bin size chosen to be $\sqrt{N}$ (see Stanley Chan, Chapter 3.2.5).

   (e) For Mode 2, (1) print the number of times 2 appears in the the samples of X generated from uniform samples and (2) plot a histogram of the samples of X using bin size to be $\sqrt{N}$.

2.2 Grading

This question accounts to 30 marks with a split of 5+10+15 for the three portions.

3 Generating equally likely k subsets of n

Given, n, k, N, generate $n \times N$ uniform $[0, 1]$ samples. $U_{i,j}$ are the samples where $i = 1, 2, \cdots, n$ and $j = 1, 2, \cdots, N$. For each of the n uniform samples you are expected to generate a k subset. Therefore you will be generating N subsets. Each subset can be representing using n indicator values that are determined from the uniform samples as shown below.

$$I_{1,j} = \begin{cases} 1 & U_{1,j} \le \frac{k}{n} \\ 0 & \text{otherwise} \end{cases}$$

Once $I_{1,j}$, $I_{2,j} \cdots, I_{i,j}$ are determined, get $I_{i+1,j}$ by setting:

$$I_{i+1,j} = \begin{cases} 1 & U_{i+1,j} \le \frac{k - \sum_{u=1}^{i} I_{u,j}}{n-i} \\ 0 & \text{otherwise.} \end{cases}$$

To see the proof for why this results in equally likely subsets, please refer to Chapter 6 of Sheldon M. Ross.

Using this you are able to generate N subsets of size k. Suppose $n = 3, k = 2$ and $I_{1,1} = 0, I_{2,1} = 1, I_{2,2} = 1$ then the first subset sample is $\{2, 3\}$. Each subset can be represented using a decimal number in $[0 : 2^n - 1]$. For example, $I_{1,1} = 0, I_{2,1} = 1, I_{2,2} = 1$ can be represented using decimal number 6 as convert_binary_to_dec$(110) = 6$

$$\begin{array}{|c|c|c|c|c|} \hline 0.8147 & 0.9134 & 0.2785 & 0.9649 & 0.9572 \\ 0.9058 & 0.6324 & 0.5469 & 0.1576 & 0.4854 \\ 0.1270 & 0.0975 & 0.9575 & 0.9706 & 0.8003 \\ \hline \end{array} \rightarrow \begin{array}{|c|c|c|c|c|} \hline 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ \hline \end{array} \rightarrow \begin{array}{|c|c|c|c|c|} \hline 6 & 6 & 3 & 6 & 6 \\ \hline \end{array} \qquad (1)$$

Shown above is example for $n = 3, k = 2, N = 5$ where $n \times N$ uniform samples $\{U_{i,j}\}$ are converted to $n \times N$ binary values $\{I_{i,j}\}$ and then to N subset values represented as a decimal number in $[0 : 2^n - 1 = 7]$. Note that we only see binary value with weight $k = 2$ as we are picking subsets of size 2. In this case we are expected to see subsets $\{1, 2\}, \{1, 3\}, \{2, 3\}$ that correspond to binary vectors $(0, 1, 1), (1, 0, 1), (1, 1, 0)$ and decimal numbers $3, 5, 6$.

3.1 Requirements

1. Input requirements: Python file should accept input arguments in the following form:
   (a) "python 3_Myna.py 3 2 100" should run for the case $n = 3, k = 2, N = 100$.
   (b) "python 3_Myna.py 4 2 10000" should run for the case $n = 4, k = 2, N = 10000$.
2. Output requirements: You are expected to plot histogram of the subset values (i.e., the decimal values that correspond to them). For $n = 3, k = 2, N = 5$ example shown in equation (1) the histogram will have values $(0, 0, 0, 1, 0, 0, 4, 0)$ corresponding to subset indices $(0, 1, 2, 3, 4, 5, 6, 7)$. As N increases it is expected that histogram values get closer to $(0, 0, 0, N/3, 0, N/3, N/3, 0)$ i.e., all the possible two subsets are equally likely.

3.2 Grading

This question carries 20 marks.

4 Bertrand's Paradox

Bertrand's paradox problem is usually presented in the first lecture of a probability course to motivate why one needs to formally define probability model through sample space, event space and the probability rule trio $(\Omega, \mathcal{F}, P)$.

Suppose one is interested in finding the probability that a randomly chosen chord of a unit radius circle has length greater than that of the side of an equilateral triangle embedded in it. Depending of how the experiment of choosing a chord is done, the answer could be either 1/3 or 1/2 or 1/4. As part of this assignment you will be simulating all these three experiments.

Assume that the unit radius circle is centered at $(0, 0)$.

4.1 Mode 0: Angle made by chord w.r.t a tangent at one fixed end is equally likely

In this method, it is assumed that the end point of the chord is equally likely to be anywhere on the perimeter of the circle. So without loss of generality we will assume one end point is at $(0, 1)$. The experiment is done by picking angle the chord makes with tangent passing through $(0, 1)$ to be equally likely.

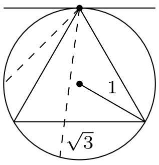


Figure 1: The dashed line is a chord that makes an angle Θ with tangent passing through the top of the circle. It can be seen that the chord has length $\geq \sqrt{3}$ iff $\Theta \in [60 : 120]$ from the figure above.

Let X be the random variable representing the chord length. Determine it as a function of Θ from the geometry described above. Now generate N uniform random samples of Θ in $[0 : \pi]$ and use them to generate samples for chord length. In this case, the probability that chord length is greater than $\sqrt{3}$ is $P(X \geq \sqrt{3}) = P(\Theta \in [\pi/3 : 2\pi/3]) = 1/3$.

4.1.1 Output requirements

1. Print the fraction of the chord length samples that are greater than $\sqrt{3}$.
2. Plot the histogram of the chord length samples. Use the square root rule to choose the bin size i.e., number of bins is $\sqrt{N}$.

4.2 Mode 1: Distance of the chord from center is equally likely

In this method, it is assumed that the distance from the center of the chord to the circle center is equally like to be in $[0, 1]$. Assuming the angle made by the chord with x-axis is equally likely to be $[0 : \pi]$, we consider that the chord is parallel to x-axis and appears below the x-axis without loss of generality.

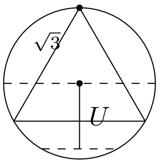


Figure 2: The dashed line is the chord is at distance U from the center. $U \sim \text{Uniform}[0, 1]$.

Determine the chord length Y as function of U in this case. Now generate N uniform random samples of U in $[0 : 1]$ and use them to generate samples of Y for chord length. In this case, the probability that chord length is greater than $\sqrt{3}$ is $P(Y \geq \sqrt{3}) = P(U \in [0 : 1/2]) = 1/2$.

4.2.1 Output requirements

1. Print the fraction of the chord length samples that are greater than $\sqrt{3}$.
2. Plot the histogram of the chord length samples. Use the square root rule to choose the bin size i.e., number of bins is $\sqrt{N}$.

4.3 Mode 2: Center of the chord is equally likely within circle

Here it is assumed that the center of the chord is equally likely to be a point in unit circle. From the description of the previous mode, we know that if distance of the chord center from the center of circle is $\leq 1/2$, then the length of chord is $\geq \sqrt{3}$. Here, the probability that the center of chord is at a distance $\leq 1/2$ from circle center is given by the area of smaller circle with radius 1/2 i.e., the probability is 1/4 in this case.

Given (X, Y) is uniformly random within a circle of radius 1 centered at $(0, 0)$. The chord length is determined by $\sqrt{X^2 + Y^2}$ which is the distance of chord's center from the circle center. Find the CDF of $R = \sqrt{X^2 + Y^2}$. Use CDF function $F_R(r)$ together with the fact that $F_R^{-1}(U)$ is a random variable with CDF $F_R(r)$ given that U is uniform random to generate samples of R. Now determine chord length Z as a function of R, the distance of chord from the circle center. Apply this to get the chord length samples i.e., samples of Z.

4.3.1 Output requirements

1. Print the fraction of the chord length samples that are greater than $\sqrt{3}$.
2. Plot the histogram of the chord length samples. Use the square root rule to choose the bin size i.e., number of bins is $\sqrt{N}$.

4.4 Requirements

1. Input requirements
   (a) first argument indicates mode. It can take values $0, 1, 2$.
   (b) second argument is the number of samples N
2. Example commands that you should test with:
   (a) `python 2_Myna.py 0 1000`
   (b) `python 2_Myna.py 1 1000`
   (c) `python 2_Myna.py 2 1000`

   Increase N to validate that the probability of chord length exceeding $\sqrt{3}$ matches with 1/3, 1/2, 1/4 for modes 0, 1, 2 respectively.

4.5 Grading

This question counts to 35 marks with split of 10+10+15 across the three modes.

5 References

1. https://algebra.math.bme.hu/2019-20-1/BMETE91AM46-A1#11
2. A first course on probability, Sheldon M. Ross, 10th edition.
3. Stanley Chan, Introduction to Probability for DATA SCIENCE