EE1080/AI1110/EE2102 Probability and Random Processes

15th April, 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 Central Limit Theorem

In this problem, you'll have to first:

1. generate $N \times n$ samples of a Bernoulli(p) random variable or;
2. generate $N \times n$ samples of a Uniform($[0, 1]$) random variable or;

3. generate $N \times n$ samples of an exponential(λ) samples.

You can use available python libraries to generate these samples. Let $X_{i,j}$ be the sample at i-th row and j-th column. Collect the row averages to a vector Y_i where:

$$Y_i = \frac{\sum_{j=1}^{n} X_{i,j}}{n}$$

Plot the histogram corresponding to the N averages along with the pdf of normal distribution with mean μ and σ^2/n on the same figure. μ and σ^2 are mean and variance of $X_{i,j}$ respectively. *Scale the histogram so that the visually pdf and histogram can be compared. Try out normed=1 for histogram.*

1.1 Requirements

1. Input arguments to the this program:
   (a) first argument is mode it indicates, if it is 0, then your program should generate Bernoulli samples, if 1, Uniform samples and if 2, exponential samples.
   (b) second and third arguments are n and N respectively
   (c) fourth argument is the parameter of the distribution i.e., if mode is 0 it is p, if mode is 2 it is λ.
2. Example commands that you should test with:
   (a) `python 1_Myna.py 0 100 10000 0.5`
   (b) `python 1_Myna.py 0 5 10000 0.5`
   (c) `python 1_Myna.py 1 100 10000`
   (d) `python 1_Myna.py 1 5 10000`
   (e) `python 1_Myna.py 2 100 10000 5`
   (f) `python 1_Myna.py 2 5 10000 5`
3. Output of this program.
   (a) Single figure with two plots, one that corresponds to histogram of sample means and other with PDF of the Gaussian R.V with mean μ and variance σ^2/n computed from the parameters of the distribution (p, λ depending on the mode) and n.

1.2 Grading

This question accounts to 20 marks with equal split for the three modes.

2 Ordered Statistics

Puchku organizes a party for 6pm, inviting 6 friends. She loves to play Dungeons and Dragons that requires at least 4 players (besides her for it to be fun). Everyone is excited, so as soon as the 4th friend has arrived, she is starting the game. Her friends arrive in a uniform distribution between 6 and 7pm (and no one arrives before 6, but everybody arrives until 7pm). Let X denote the ratio of the starting time in the interval $[6, 7]$ (ie. X is a random variable with value between 0 and 1 i.e., if the fourth friend arrived at 6.30pm then $X = 0.5$). Write a function simulating X. "The idea here is that to generate N samples of X, generate $6N$ uniform $[0, 1]$ samples and place them in an $N \times 6$ matrix. Sort each row of the matrix in ascending order and pick the fourth column in the sorted matrix that correspond to N samples of X."

Consider the following two density functions:

$$f_a(x) = \frac{6!}{3!2!}x^3(1-x)^2, \quad 0 \le x \le 1,$$
$$f_b(x) = \frac{6!}{4!1!}x^4(1-x), \quad 0 \le x \le 1.$$

1. Input arguments to the this program:
   (a) first argument is number of samples N
2. Example commands that you should test with:

(a) python 2_Myna.py 10000

3. Output of this program.

   (a) Plot the histogram of samples of X. Consider 100 bins of size 0.01 each. Also plot the two pdfs shown above in the same figure. Use normed=1 for histogram plot so that it can be compared with the pdfs.

   (b) Print a if the histogram looks close to f_a otherwise print f_b.

2.1 Grading

This question accounts to 20 marks.

3 MMSE Estimation (Maximal Ratio Combining)

Let $X, W_1, W_2, \cdots, W_n$ be i.i.d normal random variables. X is normal random variable with mean μ_0 and variance σ_0^2 and W_i is normal random variable with mean $\mu_i = 0$ and variance σ_i^2. Let:

$$Y_i = X + W_i$$

for all $i = 1, 2, \cdots, n$. You are given $Y_1, Y_2, \cdots, Y_n$ are expected to estimate X from them. Find the MMSE estimator $E[X|Y_1, Y_2, \cdots, Y_n]$ for this problem in terms of $Y_1, Y_2, \cdots, Y_n$ and the problem parameters μ_i, σ_i^2 for $i = 0, 1, 2, \cdots, n$. Note here that:

$$\begin{aligned} f_{Y_i|X}(y_i|x) &= f_{W_i}(y_i - x) \quad \text{(check why this is true)} \\ f_{Y_1,\cdots,Y_n|X}(y_1,\cdots,y_n|x) &= \prod_{j=1}^{n} f_{Y_j|X}(y_j|x) \text{(check why this is true)} \\ f_X(x) &= \frac{1}{\sqrt{2\pi\sigma_0^2}} e^{-\frac{1}{2\sigma_0^2}(x-\mu_0)^2}, \quad f_{W_i}(w) = \frac{1}{\sqrt{2\pi\sigma_i^2}} e^{-\frac{1}{2\sigma_i^2}(w)^2} \end{aligned}$$

Use them to find $f_{X|Y_1,Y_2,\cdots,Y_n}(x|y_1,\cdots,y_n)$ and $E[X|Y_1, Y_2, \cdots, Y_n]$ (refer to Chapter 8 of Reference 3). You can use the lecture notes material to solve for linear MMSE estimate using n measurements for this problem as the linear estimator will match with MMSE estimator for the joint Gaussian case.

1. Input arguments to the this program:

   (a) first argument and second arguments are μ_0 and σ_0^2 respectively.

   (b) third argument is the csv file name. This file has three ~~three~~ two columns. The first column containing the samples, second ~~containing corresponding μ_i's and third~~ containing corresponding σ_i^2 values.

2. Example commands that you should test with:

   (a) python 3_Myna.py 1 10 samples.csv

   (b) python 3_Myna.py 2 100 samples.csv

3. Output of this program.

   (a) Fetch the total number of available samples N from the samples.csv file. This should be the number of rows of the csv file.

   (b) Vary n in the set $1, 2, \cdots, N$ and get MMSE estimates, $\hat{x}(Y_1), \hat{x}(Y_1, Y_2), \cdots, \hat{x}(Y_1, Y_2, \cdots, Y_N)$. Plot these N MMSE estimates of X as scatter plot. The X-axis here are values $1, 2, \cdots, N$ and the Y-axis corresponds to the estimates. Do you see the estimates getting refined with increase in n ?

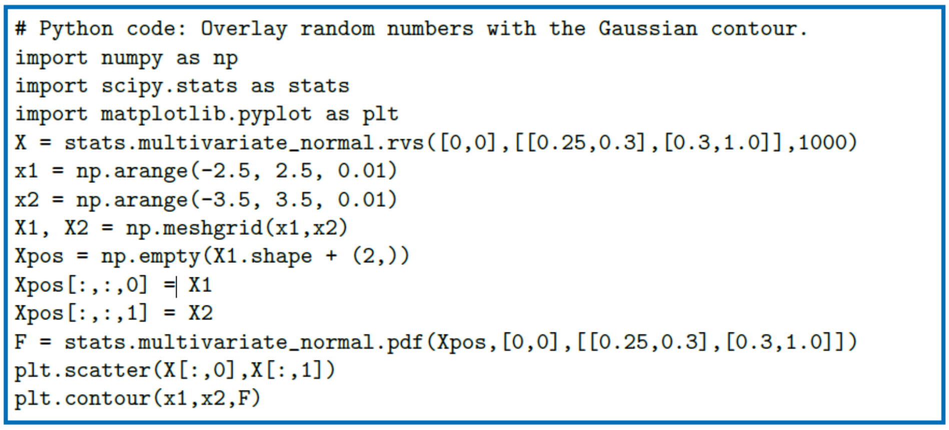
3.1 Grading

This question accounts to 30 marks ~~with equal split for the three modes.~~

4 Two Ways of Generating Jointly Gaussian Vectors

In this problem you are expected to generated samples of size 2 Gaussian vectors in two ways. The first one is by using standard libraries of python (see reference 2, chapter 5.6) where you provide the mean of Gaussian vector and the Covariance matrix and the second one is by generating 2 independent standard normal Gaussian variables and transforming them to get the Gaussian vectors of required mean and variance.

1. Input requirements: There are 7 inputs to this program
   (a) First two inputs correspond to mean values of two random variables
   (b) Next four inputs correspond to the entries of the covariance matrix of the Gaussian vector X
   (c) Last input is the number of samples N.
2. Output requirements
   (a) Print the mean 2×1 vector and covariance matrix K of size 2×2 as read from the input arguments.
   (b) Have checks in your program to verify that the input covariance matrix, K is symmetric and is positive semi-definite (do eigen value decomposition, check eigen values are all ≥ 0).
   (c) Generate a figure with scatter plots of the N samples generated from standard python libraries to generate multi normal random variable functions. (see reference 2, chapter 5.6)
   (d) Generating samples form iid standard normal random variables.
      i. Use the covariance matrix K to do a orthogonal diagonalization of the form: $K = UDU^T$ where $UU^T = I$. (there should be standard python functions available for you to do this, assumption is that D has eigen vectors in decreasing order that is the first diagonal element is the larget eigen value and the first column of U is the eigen vector corresponding to it.)
      ii. Set $A = UD^{1/2}$. Generate $2 \times N$ samples of standard normal random variables, represented by matrix S. $AS + M$ results in N Gaussian vectors with the required mean M, and covariance K as provided.
   (e) In both the figures that contain the samples, draw a contour (see image below for code snippet) that represents the joint pdf of multinormal distribution through the contour lines where each ellipse represents points with equal density.



```
# Python code: Overlay random numbers with the Gaussian contour.
import numpy as np
import scipy.stats as stats
import matplotlib.pyplot as plt
X = stats.multivariate_normal.rvs([0,0],[[0.25,0.3],[0.3,1.0]],1000)
x1 = np.arange(-2.5, 2.5, 0.01)
x2 = np.arange(-3.5, 3.5, 0.01)
X1, X2 = np.meshgrid(x1,x2)
Xpos = np.empty(X1.shape + (2,))
Xpos[:,:,0] = X1
Xpos[:,:,1] = X2
F = stats.multivariate_normal.pdf(Xpos,[0,0],[[0.25,0.3],[0.3,1.0]])
plt.scatter(X[:,0],X[:,1])
plt.contour(x1,x2,F)
```

Figure 1: Code snippet from reference 2, Chapter 5.6 by Stanley Chan to generate contour plots of joint pdf. Note that the ranges of x_1 and x_2 are picked to be symmetric around $(0, 0)$. Suggestion for you is to pick them symmetric with respect to $M = (m_1, m_2)$

   (f) Draw a line from the point $M = (m_1, m_2)$ in the direction of vector $\underline{u}_1$ which is given by first column of U and also towards vector $\underline{u}_2$ which is given by the second column of U. Do you see the contour looking like an ellipse tilted in the direction of $\underline{u}_1$, $\underline{u}_2$.

4.1 Grading

This question accounts to 30 marks. ~~with equal split for the three modes.~~

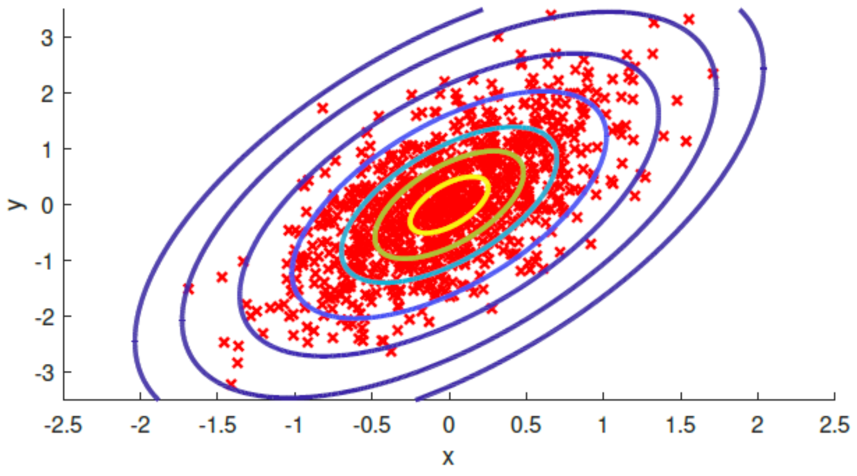


Figure 2: Figure from reference 2, Chapter 5.6 by Stanley Chan that plots samples of Gaussian vectors with mean $(0,0)$ and covariance $K = \begin{bmatrix} 0.25 & 0.3 \\ 0.3 & 1 \end{bmatrix}$. The scale of X-axis is different from that of Y-axis here. The eigen vectors $\underline{u}_1, \underline{u}_2$ for this example should be in directions 70, -19 degrees approximately.

5 References

1. `https://algebra.math.bme.hu/2019-20-1/BMETE91AM46-A1#11`
2. Stanley Chan, Introduction to Probability for DATA SCIENCE
3. Introduction to Probability by Bertsekas and Tsitsiklis