Random Variables in Python

G V V Sharma*

Contents

***Abstract*—This manual provides a simple introduction to elementary probability and random variables. This is done by generating random variables in Python and computing metrics like the CDF and PDF for some random variables. In the process, basic concepts like hypothesis testing, transformation of random variables, central limit theorem, etc.. are introduced.**

1 Preliminaries

1.1 The Gaussian Distribution

Problem 1.1. Generate a Gaussian random number with 0 mean and unit variance.

*The author is with the Department of Electrical Engineering, Indian Institute of Technology, Hyderabad 502205 India e-mail: gadepall@iith.ac.in.

Solution: Open a text editor and type the following program.

```
#!/usr/bin/env python

#This program generates a Gaussian
   random no with 0 mean and unit
  variance

#Importing numy, scipy, mpmath and
   pyplot
import numpy as np
import mpmath as mp
import scipy
import scipy.stats as sp
import matplotlib.pyplot as plt
import subprocess

print np.random.normal(0,1)
```

Save the file as gaussian_no.py and run the program.

Problem 1.2. The mean of a random variable X is defined as

$$E[X] = \frac{1}{N}\sum_{i=1}^{N} X_i \tag{1.1}$$

and its variance as

$$\text{var}[X] = E[X - E[X]]^2 \tag{1.2}$$

Verify that the program in 1.1 actually generates a Gaussian random variable with 0 mean and unit variance.

Solution: Use the header in the previous program, type the following code and execute.

```
#!/usr/bin/env python

#This program generates a Gaussian
   random no with 0 mean and unit
  variance
```

```
#Importing numy, scipy, mpmath and
    pyplot
import numpy as np
import mpmath as mp
import scipy
import scipy.stats as sp
import matplotlib.pyplot as plt
import subprocess



simlen = 1e5 #No of samples

n = np.random.normal(0,1,simlen)#
    Random vector

mean = np.sum(n)/simlen #Mean
    value

print mean

var = np.sum(np.square(n - mean*np
    .ones((1,simlen))))/simlen

print var
```

Problem 1.3. Using the previous program, verify you results for different values of the mean and variance.

1.2 CDF and PDF

Problem 1.4. A Gaussian random variable X with mean 0 and unit variance can be expressed as $X \sim \mathcal{N}(0, 1)$. Its cumulative distribution function (CDF) is defined as

$$F_X(x) = \Pr(X < x), \tag{1.3}$$

Plot $F_X(x)$.

Solution: The following code yields Fig. 1.4.

```
#!/usr/bin/env python



#Importing numy, scipy, mpmath and
    pyplot
import numpy as np
import mpmath as mp
import scipy
import scipy.stats as sp
#from scipy.integrate import quad
#import scipy.integrate as spint
import matplotlib.pyplot as plt
import subprocess



x = np.linspace(-4,4,30)#points on
    the x axis
simlen = 1e5 #number of samples
err = [] #declaring probability
    list
n = np.random.normal(0,1,simlen)

for i in range(0,30):
        err_ind = np.nonzero(n < x
            [i]) #checking
            probability condition
        err_n = np.size(err_ind) #
            computing the
            probability
        err.append(err_n/simlen) #
            storing the probability
            values in a list


plt.plot(x.T,err)#plotting the CDF
plt.grid() #creating the grid
plt.xlabel('$x$')
plt.ylabel('$F_X(x)$')
plt.show() #opening the plot
    window
```

Problem 1.5. List the properties of $F_X(x)$ based on Fig. 1.4.

Problem 1.6. Let

$$p_X(x_i) = \frac{F_X(x_i) - F_X(x_{i-1})}{h}, i = 1, 2, \dots h \tag{1.4}$$

for $x_i = x_{i-1} + h, x_1 = -4$. Plot $p_X(x_i)$. On the same graph, plot

$$p_X(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}, -4 < x < 4 \tag{1.5}$$

Solution: The following code yields the graph in Fig. 1.6

```
#!/usr/bin/env python
```

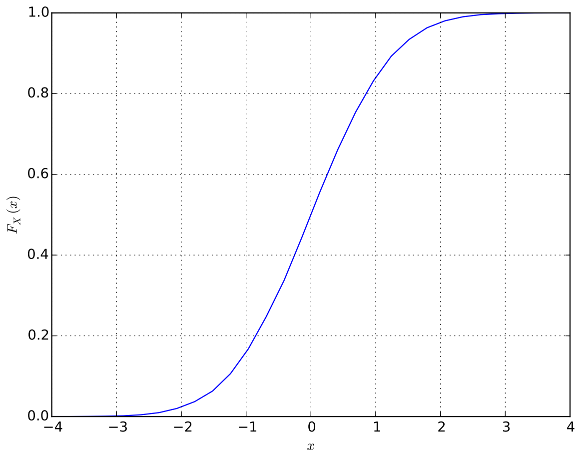

Fig. 1.4: CDF of X

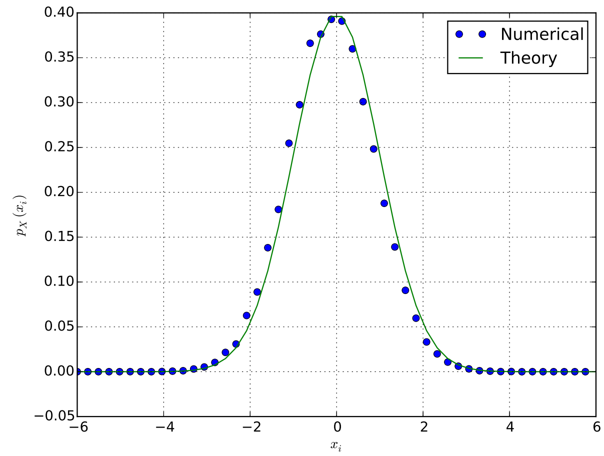

Fig. 1.6: The PDF of X

```
#Importing numy, scipy, mpmath and
    pyplot
import numpy as np
import mpmath as mp
import scipy
import scipy.stats as sp
#from scipy.integrate import quad
#import scipy.integrate as spint
import matplotlib.pyplot as plt
import subprocess


maxrange=50
maxlim=6.0
x = np.linspace(-maxlim,maxlim,
   maxrange)#points on the x axis
simlen = 1e5 #number of samples
err = [] #declaring probability
   list
pdf = [] #declaring pdf list
h = 2*maxlim/(maxrange-1);
n = np.random.normal(0,1,simlen)

for i in range(0,maxrange):
        err_ind = np.nonzero(n < x
           [i]) #checking
           probability condition
        err_n = np.size(err_ind) #
           computing the
           probability
        err.append(err_n/simlen) #
           storing the probability
           values in a list


for i in range(0,maxrange-1):
        test = (err[i+1]-err[i])/(
           x[i+1]-x[i])
        pdf.append(test) #storing
           the pdf values in a list

def gauss_pdf(x):
        return 1/np.sqrt(2*np.pi)*
           np.exp(-x**2/2.0)

vec_gauss_pdf = scipy.vectorize(
   gauss_pdf)

plt.plot(x[0:(maxrange-1)].T,pdf,'
   o')
plt.plot(x,vec_gauss_pdf(x))#
   plotting the CDF
plt.grid() #creating the grid
plt.xlabel('$x_i$')
plt.ylabel('$p_X(x_i)$')
plt.legend(["Numerical","Theory"])
plt.show() #opening the plot
   window
```

Thus, the PDF is the derivative of the CDF. For $X \sim \mathcal{N}(0,1)$, the PDF is

$$p_X(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}, \quad -\infty < x < \infty \tag{1.6}$$

Problem 1.7. For $X \sim \mathcal{N}\left(\mu, \sigma^2\right)$,

$$p_X(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \quad -\infty < x < \infty \tag{1.7}$$

Plot $p_X(x)$ for different values of μ and σ in the same graph. Comment.

2 Hypothesis

2.1 Detection & Estimation

Problem 2.1. Let $X \in \{1, -1\}$. Generate X such that the numbers 1 and -1 appear with equal probability. This is a random variable formulation of the coin tossing experiment.

Solution: The following script generates the numbers 1 and -1 with equal probability.

```
#!/usr/bin/env python



#Importing numy, scipy, mpmath and
    pyplot
import numpy as np
import mpmath as mp
import scipy
import scipy.stats as sp
#from scipy.integrate import quad
#import scipy.integrate as spint
import matplotlib.pyplot as plt
import subprocess

#Function for generating coin toss
def coin(x):
        return 2*np.random.randint
            (2,size=x)-1




print coin(1)
```

Problem 2.2. Verify that the script in the previous problem generates equiprobable symbols.

Problem 2.3. Suppose $X \in \{1, -1\}$ and

$$Y = AX + N \tag{2.1}$$

where $N \sim \mathcal{N}(0, 1)$ and $A = 4$. Plot Y.

Solution: The following code yields Fig. 2.3

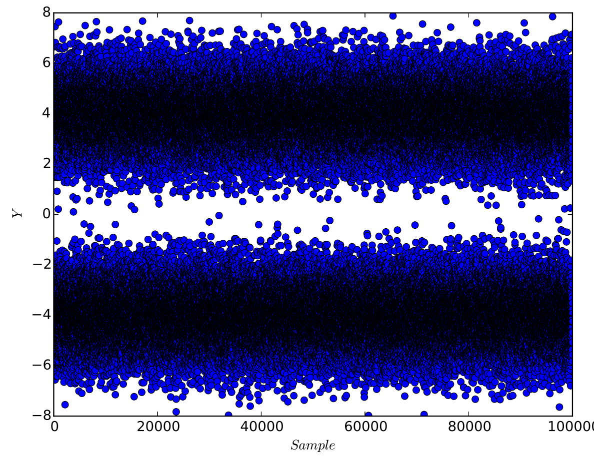

Fig. 2.3: Plot of Y

```
#!/usr/bin/env python



#Importing numy, scipy, mpmath and
    pyplot
import numpy as np
import mpmath as mp
import scipy
import scipy.stats as sp
#from scipy.integrate import quad
#import scipy.integrate as spint
import matplotlib.pyplot as plt
import subprocess

#Function for generating coin toss
def coin(x):
        return 2*np.random.randint
            (2,size=x)-1



simlen = 1e5
N = np.random.normal(0,1,simlen)
X = coin(simlen)
A = 4
Y = A*X+N

plt.plot(Y,'o')
plt.xlabel('$Sample}$')
plt.ylabel('$Y$')
plt.show()
```

Problem 2.4. Given Y in the previous problem, how would you decide whether X is 1 or -1.

Problem 2.5. Suppose $X = 1$ and $\hat{X}$ is what you detected. Find $\Pr\left(\hat{X} = -1/X = 1\right)$.

Problem 2.6. Plot $\Pr\left(\hat{X} = -1/X = 1\right)$ with respect to A.

Problem 2.7. For $X \sim \mathcal{N}(0, 1)$, the Q-function is defined as

$$Q(x) = \Pr(X > x), \quad x > 0 \tag{2.2}$$

Express $\Pr\left(\hat{X} = -1/X = 1\right)$ in terms of the Q-function. Plot this expression with respect to A and compare with the result obtained through simulation.

Problem 2.8. The signal to noise ratio of the above system is defined as

$$SNR = \frac{A^2}{E[N^2]} \tag{2.3}$$

Plot the thoeretical and simulated values of $\Pr\left(\hat{X} = -1/X = 1\right)$ for the SNR ranging from 0 to 10 dB.

Problem 2.9. Now consider a threshold $\lambda > 0$ and find the average probability of error. Plot this with respect to λ.

Problem 2.10. From the graph in the previous problem, find the optimum threshold so that the probability of error is minimum.

2.2 *The MAP criterion*

A Gaussian random variable $Y \sim \mathcal{N}\left(\mu, \sigma^2\right)$ has the pdf

$$p_Y(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(x-\mu)^2/2\sigma^2} \quad -\infty < x < \infty \tag{2.4}$$

Problem 2.11. Plot

$$p_Y(Y|X = 1) \text{ and } p_Y(Y|X = -1) \tag{2.5}$$

in the same graph with respect to A.

Problem 2.12. Graphically obtain the decision resulting from

$$p_Y(Y|X = 1) \underset{-1}{\overset{1}{\gtrless}} p_Y(Y|X = -1) \tag{2.6}$$

Comment.

3 Transformation of Variables

3.1 *Using Definition*

Problem 3.1. Let $X_1 \sim \mathcal{N}(0, 1)$ and $X_2 \sim \mathcal{N}(0, 1)$. Plot the CDF and PDF of

$$V = X_1^2 + X_2^2 \tag{3.1}$$

Problem 3.2. If

$$F_V(x) = \begin{cases} 1 - e^{-\alpha x} & x \geq 0 \\ 0 & x < 0, \end{cases} \tag{3.2}$$

find α.

Problem 3.3. Plot the CDF and PDf of

$$A = \sqrt{V} \tag{3.3}$$

Problem 3.4. Find an expression for $F_A(x)$ using the definition. Plot this expression and compare with the result of problem 3.3.

Problem 3.5. Find an expression for $p_A(x)$.

3.2 *Using Jacobian*

Problem 3.6. Evaluate the joint PDF of X_1, X_2, given by

$$p_{X_1,X_2}(x_1, x_2) = p_{X_1}(x_1)\, p_{X_2}(x_2) \tag{3.4}$$

Problem 3.7. Let

$$X_1 = \sqrt{V}\cos\theta \tag{3.5}$$
$$X_2 = \sqrt{V}\sin\theta. \tag{3.6}$$

Evaluate the Jacobian

$$J = \begin{vmatrix} \frac{\partial x_1}{\partial v} & \frac{\partial x_2}{\partial v} \\ \frac{\partial x_1}{\partial \theta} & \frac{\partial x_2}{\partial \theta} \end{vmatrix} \tag{3.7}$$

Problem 3.8. Find

$$p_{V,\Theta}(v, \theta) = |J|\, p_{X_1,X_2}(x_1, x_2) \tag{3.8}$$

Problem 3.9. Find $p_V(v)$.

Problem 3.10. Find $p_\Theta(\theta)$.

Problem 3.11. Are Y and Θ independent?

Problem 3.12. Find $p_A(x)$ using the Jacobian.

4 Conditional Probability

Problem 4.1. Plot

$$P_e = \Pr\left(\hat{X} = -1|X = 1\right) \tag{4.1}$$

for

$$Y = AX + N, \tag{4.2}$$

where A is Raleigh with $E\left[A^2\right] = \gamma, N \sim \mathcal{N}(0,1), X \in (-1,1)$ for $0 \le \gamma \le 10$ dB.

Problem 4.2. Assuming that N is a constant, find an expression for P_e. Call this $P_e(N)$

Problem 4.3. For a function g,

$$E\left[g(X)\right] = \int_{-\infty}^{\infty} g(x) p_X(x)\, dx \tag{4.3}$$

Find $P_e = E\left[P_e(N)\right]$.

Problem 4.4. Plot P_e in problems 4.1 and 4.3 on the same graph w.r.t γ. Comment.

5 Two Dimensions

Let

$$\mathbf{y} = A\mathbf{x} + \mathbf{n}, \tag{5.1}$$

where

$$x \in (\mathbf{s}_0, \mathbf{s}_1), \mathbf{s}_0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \mathbf{s}_1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{5.2}$$

$$\mathbf{n} = \begin{pmatrix} n_1 \\ n_2 \end{pmatrix}, n_1, n_2 \sim \mathcal{N}(0,1). \tag{5.3}$$

Problem 5.1. Plot

$$\mathbf{y}|\mathbf{s}_0 \text{ and } \mathbf{y}|\mathbf{s}_1 \tag{5.4}$$

on the same graph using a scatter plot.

Problem 5.2. For the above problem, find a decision rule for detecting the symbols $\mathbf{s}_0$ and $\mathbf{s}_1$.

Problem 5.3. Plot

$$P_e = \Pr\left(\hat{\mathbf{x}} = \mathbf{s}_1 | \mathbf{x} = \mathbf{s}_0\right) \tag{5.5}$$

with respect to the SNR from 0 to 10 dB.

Problem 5.4. Obtain an expression for P_e. Verify this by comparing the theory and simulation plots on the same graph.

6 Transform Domain

Let $X \sim \mathcal{N}\left(\mu, \sigma^2\right)$.

Problem 6.1. Find $M_X(s) = E\left[e^{-sX}\right]$.

Problem 6.2. Let

$$N = n_1 - n_2, \quad n_1, n_2 \sim \mathcal{N}(0,1). \tag{6.1}$$

Find $M_N(s)$, assuming that n_1 and n_2 are independent.

Problem 6.3. Show that N is Gaussian. Find its mean and variance. Comment.

7 Random Number Generation

7.1 Uniform Random Numbers

Let U be a uniform random variable between 0 and 1.

Problem 7.1. Generate 10^6 samples of U using a C program and save into a file called uni.dat .

Problem 7.2. Load the uni.dat file into python and plot the empirical CDF of U using the samples in uni.dat.

Problem 7.3. Verify that your CDF in the above problem is correct by plotting the theoretical $F_U(x)$.

7.2 Central Limit Theorem

Problem 7.4. Generate $U_i, i = 1, 2, \ldots, 12$, a set of independent uniform random variables between 0 and 1 using a C program.

Problem 7.5. Generate 10^6 samples of the random variable

$$S = \sum_{i=1}^{12} U_i - 6 \tag{7.1}$$

and save in a file called s.dat

Problem 7.6. Load s.dat in python and plot the empirical PDF of S using the samples in s.dat. Does it look familiar? Comment.

7.3 From Uniform to Other

Problem 7.7. Generate samples of

$$V = -2\ln(1 - U) \tag{7.2}$$

and plot its CDF. Comment.

Problem 7.8. Generate the Rayleigh distribution from Uniform. Verify your result through graphical plots.