Linear Classification

G V V Sharma*

Contents

***Abstract*—This manual provides an introduction to linear lethods in regression.**

1 The Gaussian Distribution

1.1 Generate a Gaussian random number with 0 mean and unit variance.
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 numpy
import numpy as np

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

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

*The author is with the Department of Electrical Engineering, Indian Institute of Technology, Hyderabad 502285 India e-mail: gadepall@iith.ac.in.

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.

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

#Importing numpy
import numpy as np


simlen = int(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)
```

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

2 CDF and PDF

2.1 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

$$F_X(x) = \Pr(X < x), \tag{2.1}$$

Plot $F_X(x)$.
Solution: The following code yields Fig. 2.1.

```
#Importing numpy, scipy, mpmath and pyplot
import numpy as np
import matplotlib.pyplot as plt




x = np.linspace(−4,4,30)#points on the x axis
simlen = int(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
```

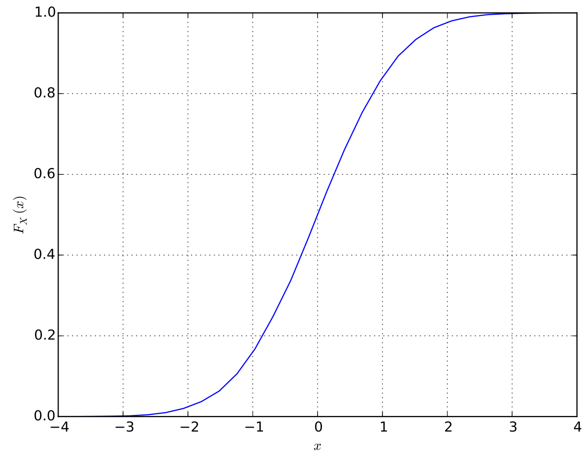

Fig. 2.1: CDF of X

2.2 List the properties of $F_X(x)$ based on Fig. 2.1.
2.3 Let

$$p_X(x_i) = \frac{F_X(x_i) - F_X(x_{i-1})}{h}, i = 1, 2, \dots h \tag{2.2}$$

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

$$p_X(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}, -4 < x < 4 \tag{2.3}$$

Solution: The following code yields the graph in Fig. 2.3

```
https://github.com/gadepall/EE1390/raw/
    master/manuals/supervised/linear_class/
    codes/1.4.py
```

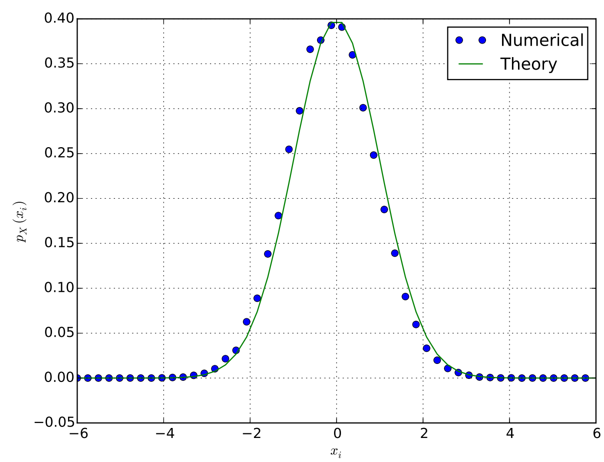

Fig. 2.3: The PDF of X

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

$$p_X(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}, \quad -\infty < x < \infty \tag{2.4}$$

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

$$p_X(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \quad -\infty < x < \infty \tag{2.5}$$

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

3 Detection & Estimation

3.1 Use the following code

```
#Importing numpy and pyplot
```

```
import numpy as np
import matplotlib.pyplot as plt


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


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

to generate X. Obtain a scatterplot of X.

3.2 Suppose you wanted to classify X into two groups. How would you do so by looking at the scatterplot?

4 Bayes Classifier

4.1 Let $(\mathbf{X}, \mathbf{G})$ be an input/output dataset, whose relation f is unknown. Also

$$\mathbf{g} \in \mathbf{G} = \{\mathbf{g}_k\}_{k=1}^{K} \tag{4.1}$$

Let

$$C(\mathbf{g}_k, \mathbf{g}_l) = \begin{cases} 1 & k = l \\ 0 & k \neq l \end{cases} \tag{4.2}$$

where $\mathbf{g}_i$ are different classes of output data. Thus C is a *correctness* metric.

4.2 Show that

$$\max_{\mathbf{g} \in \mathbf{G}} E\left[C\left(\mathbf{G}, f(\mathbf{X})\right)\right] = \max_{\mathbf{g} \in \mathbf{G}} p\left(\mathbf{g} | \mathbf{X} = \mathbf{x}\right) \tag{4.3}$$

Solution: In the above,

$$\max_{\mathbf{g} \in \mathbf{G}} E\left[C\left(\mathbf{G}, f(\mathbf{X})\right)\right]$$
$$= \max_{\mathbf{g} \in \mathbf{G}} E_{\mathbf{X}}\left[E_{\mathbf{G}}\left\{C\left(\mathbf{G}, f(\mathbf{x})\right)\right\}\right] \tag{4.4}$$

$$= \max_{\mathbf{g} \in \mathbf{G}} \sum_{k=1}^{K} C\left\{\mathbf{g}_k, \mathbf{g}\right\} p\left(\mathbf{g}_k | \mathbf{X} = \mathbf{x}\right) \tag{4.5}$$

From (4.2), the above expression simplifies to

$$\max_{\mathbf{g} \in \mathbf{G}} E\left[C\left(\mathbf{G}, f(\mathbf{X})\right)\right] = \max_{\mathbf{g} \in \mathbf{G}} p\left(\mathbf{g} | \mathbf{X} = \mathbf{x}\right) \tag{4.6}$$

5 Linear Discriminant Analysis

5.1 If

$$(X = x | G = 1) \sim \mathcal{N}(-A, 1) \tag{5.1}$$
$$(X = x | G = 1) \sim \mathcal{N}(A, 1) \tag{5.2}$$

plot $p_X(X = x | G = 0)$ and $p_X(X = x | G = 1)$ for $A = 4$.

5.2 Find

$$p_X(X = x | G = 0) \underset{1}{\overset{0}{\gtrless}} p_X(X = x | G = 1) \tag{5.3}$$

5.3 Show that

$$p_X(G = 0 | X = x) \underset{1}{\overset{0}{\gtrless}} p_X(G = 1 | X = x) \tag{5.4}$$

$$\implies p_X(X = x | G = 0) \underset{1}{\overset{0}{\gtrless}} p_X(X = x | G = 1) \tag{5.5}$$

if

$$p(G = 0) = p(G = 1) \tag{5.6}$$

6 Quadratic Discriminant Analysis

6.1 Find