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))
```

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

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

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
```

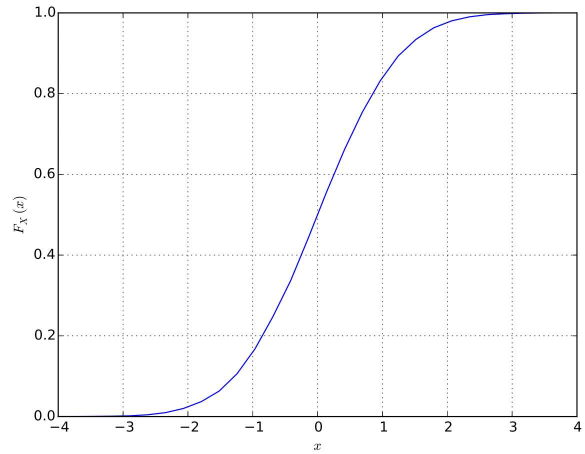

Fig. 2.1: CDF of X

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

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



maxrange=50
maxlim=6.0
x = np.linspace(−maxlim,maxlim,maxrange)#
    points on the x axis
simlen = int(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/mp.sqrt(2∗np.pi)∗np.exp(−x
```

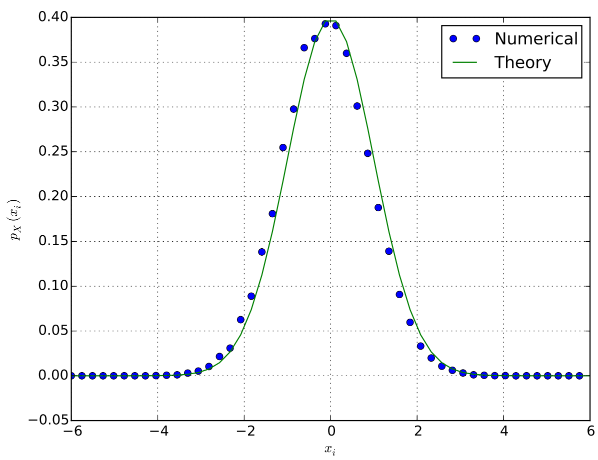

Fig. 2.3: The PDF of X

```
            **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

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

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 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.

```
#Importing numpy
import numpy as np

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



print (coin(1))
```

3.2 Verify that the script in the previous problem generates equiprobable symbols.

3.3 Suppose $X \in \{1, -1\}$ and

$$Y = AX + N \tag{3.1}$$

where $N \sim \mathcal{N}(0,1)$ and $A = 4$. Obtain a scatterplot of (X, Y).

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

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

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

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

$$Q(x) = \Pr(X > x), \quad x > 0 \tag{3.2}$$

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.

3.8 The signal to noise ratio of the above system is defined as

$$SNR = \frac{A^2}{E\left[N^2\right]} \tag{3.3}$$

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

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

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

4 The MAP criterion

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

$$p_Y(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(x-\mu)^2/2\sigma^2} - \infty < x < \infty \quad (4.1)$$

4.2 Plot

$$p_Y(Y|X = 1) \text{ and } p_Y(Y|X = -1) \quad (4.2)$$

in the same graph with respect to A.

4.3 Graphically obtain the decision resulting from

$$p_Y(Y|X = 1) \underset{-1}{\overset{1}{\gtrless}} p_Y(Y|X = -1) \quad (4.3)$$

Comment.

5 Bayes Classifier

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

$$\mathbf{g} \in \mathbf{G} = \{\mathbf{g}_k\}_{k=1}^{K} \quad (5.1)$$

Let

$$C\left(\mathbf{g}_k, \mathbf{g}_l\right) = \begin{cases} 1 & k = l \\ 0 & k \neq l \end{cases} \quad (5.2)$$

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

5.2 Show that

$$\max_{\mathbf{g}\in\mathbf{G}} E\left[C\left(\mathbf{G}, f\left(\mathbf{X}\right)\right)\right] = \max_{\mathbf{g}\in\mathbf{G}} p\left(\mathbf{g}|\mathbf{X} = \mathbf{x}\right) \quad (5.3)$$

Solution: In the above,

$$\max_{\mathbf{g}\in\mathbf{G}} E\left[C\left(\mathbf{G}, f\left(\mathbf{X}\right)\right)\right] = \max_{\mathbf{g}\in\mathbf{G}} E_{\mathbf{X}}\left[E_{\mathbf{G}}\left\{C\left(\mathbf{G}, f\left(\mathbf{x}\right)\right)\right\}\right] \quad (5.4)$$

$$= \max_{\mathbf{g}\in\mathbf{G}} \sum_{k=1}^{K} C\left\{\mathbf{g}_k, \mathbf{g}\right\} p\left(\mathbf{g}_k|\mathbf{X} = \mathbf{x}\right) \quad (5.5)$$

From (5.2), the above expression simplifies to

$$\max_{\mathbf{g}\in\mathbf{G}} E\left[C\left(\mathbf{G}, f\left(\mathbf{X}\right)\right)\right] = \max_{\mathbf{g}\in\mathbf{G}} p\left(\mathbf{g}|\mathbf{X} = \mathbf{x}\right) \quad (5.6)$$

6 Least Discriminant Analysis

6.1 Find

$$\|\mathbf{y} - \mathbf{X}\mathbf{w}\|^2 \quad (6.1)$$

Solution:

$$\|\mathbf{y} - \mathbf{X}\mathbf{w}\|^2 = \left(\mathbf{y} - \mathbf{X}\mathbf{w}\right)^T \left(\mathbf{y} - \mathbf{X}\mathbf{w}\right) \quad (6.2)$$

$$= \|\mathbf{y}\|^2 - \mathbf{w}^T\mathbf{X}^T\mathbf{y} \quad (6.3)$$

$$- \mathbf{y}^T A\mathbf{w} + \mathbf{w}^T\mathbf{X}^T\mathbf{X}\mathbf{w} \quad (6.4)$$

6.2 Assuming 2×2 matrices and 2×1 vectors, show that

$$\frac{\partial}{\partial \mathbf{w}}\mathbf{w}^T\mathbf{X}^T\mathbf{y} = \frac{\partial}{\partial \mathbf{w}}\mathbf{y}^T\mathbf{X}\mathbf{w} = \mathbf{y}^T\mathbf{X} \quad (6.5)$$

6.3 Show that

$$\frac{\partial}{\partial \mathbf{w}}\mathbf{w}^T\mathbf{X}^T\mathbf{X}\mathbf{w} = 2\mathbf{w}^T\left(\mathbf{X}^T\mathbf{X}\right) \quad (6.6)$$

6.4 Show that

$$\hat{\mathbf{w}} = \min_{\mathbf{w}} \|\mathbf{y} - \mathbf{X}\mathbf{w}\|^2 \quad (6.7)$$

$$= \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{y} \quad (6.8)$$

6.5 Using the Gram-Schmidt orthogonalization procedure, show that

$$\mathbf{X} = \mathbf{Q}\mathbf{R} \quad (6.9)$$

where $\mathbf{Q}^T\mathbf{Q} = \mathbf{I}$ and $\mathbf{R}$ is upper triangular.

6.6 Show that

$$\hat{\mathbf{w}} = \mathbf{R}\mathbf{Q}^T\mathbf{y} \quad (6.10)$$

6.7 Find $\hat{\mathbf{y}}$

7 Ridge Regression

7.1 The ridge problem is defined as

$$\hat{\mathbf{w}} = \min_{\mathbf{w}} \|\mathbf{y} - \mathbf{X}\mathbf{w}\| \quad (7.1)$$

$$\text{s.t } \|\mathbf{w}\|^2 \leq t \quad (7.2)$$

Using the Lagrangian, show that

$$\hat{\mathbf{w}} = \left(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}\right)^{-1}\mathbf{X}^T\mathbf{y} \quad (7.3)$$

8 The Lasso

8.1 The Lasso is defined as

$$\hat{\mathbf{w}} = \min_{\mathbf{w}} \|\mathbf{y} - \mathbf{X}\mathbf{w}\| \quad (8.1)$$

$$\text{s.t } \sum_{i} |w_i| \leq t \quad (8.2)$$

Obtain the corresponding Lagrangian.

8.2 Show that this is a quadratic programming problem and find a suitable algorithm.