EHB-328 İşaret İşleme için Makine Öğrenmesi

ÖDEV-2

Bilge Günsel

Ömer Bahadır Gökmen 040190030
4-11-2021

Soru.1

Iris Setosa veri kümesinden elde ettiğim kovaryans matrislerinden bulunan özdeğerler ve özvektörler:

EigenValues: [0.23374921 0.0355343 0.02631222 0.00921219]

EigenVectors: [[0.6662062 0.60592437 0.4347012 -0.00773992]
[0.7363594 -0.61961424 -0.26584753 -0.05637536]
[0.09478709 0.49163532 -0.8346381 -0.22954108]
[0.07042407 0.085021 -0.20913896 0.9716341]]

Kaynak Kod:

Verilerin Okunması:

```
def parser(data):
    x = list()
    for i in range(len(data)):
        temp = data[i].split(',')
        x.append([temp[0],temp[1],temp[2],temp[3]])
    return numpy.array(x).astype(numpy.float64)
def data_processing():
    with open('irisdata.xml') as f:
        irisdata = f.read()
    data = BeautifulSoup(irisdata, 'xml').text.split('\n')[1:-3]
    numpy.copyto(Iris,parser(data))
    numpy.copyto(Iris_setosa,Iris[:50])
    numpy.copyto(Iris_versicolor,Iris[50:100])
    numpy.copyto(Iris_virginica,Iris[100:150])
```

Kovaryans Matrisinin Hesaplanması:

```
def get_covariance_matrix(A):
    A=numpy.array(A, dtype='f')
    mean_vector=numpy.mean(A,axis=0)
    cov_matrix = numpy.reshape(numpy.zeros(Feature_number*Feature_number),
(Feature_number,Feature_number))

    for x in range(Feature_number):
        for y in range(len(A[:,x])):
            A[:,x][y]=float(A[:,x][y])-float(mean_vector[x])

    cov_matrix=(numpy.dot(numpy.transpose(A),A))/Training_number
    return cov_matrix
```

Kovaryans Matristen Öz Değerlerin ve Öz Vektörlerin Bulunması:

```
def eigvals_vectors(A):

    _matrix = get_covariance_matrix(A)
    eigvals, eigvectors = la.eig(_matrix)

    return eigvals, eigvectors
```

Tüm KOD:

```
import numpy
import csv
from bs4 import BeautifulSoup
import scipy.linalg as la
label=["sepal_length", "sepal_width", "petal_length", "petal_width", "class"]
Feature_number=4
Training_number=50
Iris_setosa= numpy.empty((50,4))
Iris_versicolor=numpy.empty((50,4))
Iris_virginica=numpy.empty((50,4))
Iris=numpy.empty((150,4))
def parser(data):
    x = list()
    for i in range(len(data)):
        temp = data[i].split(',')
        x.append([temp[0],temp[1],temp[2],temp[3]])
    return numpy.array(x).astype(numpy.float64)
```

```python
def data_processing():
    with open('irisdata.xml') as f:
        irisdata = f.read()
    data = BeautifulSoup(irisdata, 'xml').text.split('\n')[1:-3]
    numpy.copyto(Iris,parser(data))
    numpy.copyto(Iris_setosa,Iris[:50])
    numpy.copyto(Iris_versicolor,Iris[50:100])
    numpy.copyto(Iris_virginica,Iris[100:150])

def get_covariance_matrix(A):
    A=numpy.array(A, dtype='f')
    mean_vector=numpy.mean(A,axis=0)
    cov_matrix = numpy.reshape(numpy.zeros(Feature_number*Feature_number),
(Feature_number,Feature_number))

    for x in range(Feature_number):
        for y in range(len(A[:,x])):
            A[:,x][y]=float(A[:,x][y])-float(mean_vector[x])

    cov_matrix=(numpy.dot(numpy.transpose(A),A))/Training_number
    return cov_matrix

def eigvals_vectors(A):

    _matrix = get_covariance_matrix(A)
    eigvals, eigvectors = la.eig(_matrix)

    return eigvals, eigvectors

data_processing()
eigvals, eigvectors = eigvals_vectors(Iris_setosa)

print("Eigvals:\n",eigvals.real)
print("Eigvectors:\n",eigvectors)
```

Soru 2: 2 boyutlu öznitelikler kullanılarak aşağıda parametreleri verilen 2B Gauss dağılımları ile modellenen 2 örüntü sınıfını ayırd etmek için doğrusal bir Bayes sınıflandırıcı tasarlayınız. $\mu_1 = [3\ 2]^T$ $\mu_2 = [5\ 4]^T$

$$\Sigma_1 = \begin{bmatrix} .5 & \\ & .5 \end{bmatrix} \quad \Sigma_2 = \begin{bmatrix} 1 & \\ & 1 \end{bmatrix}$$

a. Sınıf 1 ve Sınıf 2 için ayırd edici fonksiyonları yazınız.
b. Bu fonksiyonları kullanarak 2B öznitelik uzayında iki sınıfa ait karar bölgelerini ayıran karar sınırının (decision boundary) denklemini yazınız.
c. Yaptığınız öğreticili (supervised) bir sınıflandırıcı tasarımı mıdır? Neden?
d. Sınıflandırıcınız doğrusal (lineer) midir? Neden?

$$g_i(x) = P(\omega_i|x) = \big(P(\omega_i)p(x|\omega_i)\big)/p(x)$$

Dağılımlar Gauss ise i. sınıfın ayırd edici fonksiyonu aşağıdaki gibi olur

$$g_i(x) = |\Sigma_i|^{-1/2} e^{-\frac{1}{2}(x-\mu_i)^T\Sigma_i^{-1}(x-\mu_i)} P(\omega_i)$$

Logaritma alınırsa i. sınıfın ayırd edici fonksiyonu aşağıdaki gibi olur

$$g_i(x) = -\frac{1}{2}(x-\mu_i)^T\Sigma_i^{-1}(x-\mu_i) - \frac{1}{2}\log|\Sigma_i| + \log P(\omega_i)$$

Ek özelliklere göre fonksiyon sadeleştirilebilir. Örn her sınıfın kovaryans matrisinin köşegen dışı elemanları sıfır ise ($\Sigma_i = \sigma_i^2 I$) fonksiyon aşağıdaki gibi sadeleşir:

$$g_i(x) = -\frac{1}{2}(x-\mu_i)^T\sigma_i^{-2}(x-\mu_i) - \frac{1}{2}N\log|\sigma_i^2| + \log P_i$$

$\mu_1 = [3\ \ 2]^T \quad \mu_2 = [5\ \ 4]^T$

$\boxed{P_1 = P_2}$

$\Sigma_1 = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix} \quad \Sigma_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

$\boxed{g_1(x) \underset{\omega_2}{\overset{\omega_1}{\gtrless}} g_2(x)}$

$$g_1(x) = -\frac{1}{2}\begin{bmatrix} x-3 \\ y-2 \end{bmatrix}^T \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} x-3 \\ y-2 \end{bmatrix} - \frac{1}{2}\log\frac{1}{4} + \log\frac{1}{2}$$

$$g_2(x) = -\frac{1}{2}\begin{bmatrix} x-5 \\ y-4 \end{bmatrix}^T \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x-5 \\ y-4 \end{bmatrix} - \frac{1}{2}\log 1 + \log\frac{1}{2}$$

a)

$$\boxed{\begin{aligned} g_1(x) &= -\left((x-3)^2 + (y-2)^2\right) - \frac{1}{2}\log\frac{1}{4} + \log\frac{1}{2} \\ g_2(x) &= -0.5\left((x-5)^2 + (y-4)^2\right) + \log\frac{1}{2} \end{aligned}}$$

$$-\left((x-3)^2 + (y-2)^2\right) - 0.69 \underset{\omega_2}{\overset{\omega_1}{\gtrless}} -0.5\left((x-5)^2 + (y-4)^2\right)$$

$$-x^2 - y^2 + 6x + \cancel{4y} - 13 - 0.69 \underset{\omega_2}{\overset{\omega_1}{\gtrless}} -0.5x^2 - 0.5y^2 + 5x + \cancel{4y} - 20.5$$

b)

$$\boxed{6.81 \underset{\omega_2}{\overset{\omega_1}{\gtrless}} 0.5x^2 + 0.5y^2 - x}$$

C.

Yaptığım tasarım önceden belirlenen karar sınırları (Decision Boundary) ile verileri sınıflandıran öğreticili (Supervised) sınıflandırmadır.

D.

Yaptığım sınıflandırma tasarımının karar sınırı karesel (Quadratic) elemanlara bağlı olduğu için doğrusal (Lineer) değildir. Karar sınırı karesel (Quadratic)bir eğri ile oluşturulmuştur.