Submission For Excercise - 2

Avik Mukherjee, Ganesh S Koparde, Chaitra Nayak

May 9, 2019

Q1. Prove the following: "If A has a finite number of distinct eigenvectors then each eigenvector must have a distinct eigenvalue."

In order to prove the above an asymmetric 2 x 2 Matrix is considered with two seperate eigen values and the condition of linear independence corresponding to the the eigen vectors of the two eigen values is derived.

Assumptions :

1. 2 x 2 Matrix(Asymmetric)

$$M = \begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{bmatrix}$$

2. Eigenvalues : λ_1, λ_2 $\lambda_1 \neq \lambda_2$

To Prove: Eigen Vector families representing values λ_1, λ_2 are linearly independent.
Proof :
Solving for:

$$M\mathbf{v} = \lambda\mathbf{v} \tag{1}$$

We get:

$$[M - \lambda I]\,\mathbf{v} = 0 \tag{2}$$

Where

λ : variable for eigen values
$\mathbf{v}$: variable representing eigen vectors corresponding to the eigen values

Solving for 2 gives:

$$\begin{bmatrix} x_{11} - \lambda & x_{12} \\ x_{21} & x_{22} - \lambda \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = 0 \tag{3}$$

Gives us a ratio between the two vector components: v_1 and v_2 as:

$$v_2 = -((x_{11} - \lambda)/x_{12})v_1 \tag{4}$$

where $\lambda = \lambda_1$ or λ_2, This leads us to two families of vectors given by:

$$\langle v_{11}, -((x_{11} - \lambda_1)/x_{12})v_{11}\rangle \, and \, \langle v_{12}, -((x_{11} - \lambda_2)/x_{12})v_{12}\rangle \tag{5}$$

where v_{11} and v_{12} represent a random scalar value. This represents two seperate families of linearly independent vectors as they only satisfy the condition for linear dependence $a\mathbf{x} + b\mathbf{y} = 0$ if a = 0 and b = 0.

Taking dimensions greater than 2 x 2 would only increase the number of equations or components for the vectors while the ratio between the vector components would completely depend on the eigen values. Thus the linear independence of the eigen vectors can be easily scaled to higher dimensions using this condition.

Q2.
a. Prove that the eingenvalues of A are positive if A is positive definite.

$$\mathbf{x}^T A\mathbf{x} \geq 0 \tag{6}$$

Considering $\mathbf{x}$ to be a eigenvector of A having eigenvalue λ_1 Equation 6 can be considered to be written as:

$$\lambda_1 \mathbf{x}^T \mathbf{x} \geq 0 \tag{7}$$

where $\mathbf{x}^T\mathbf{x} \geq 0 \; \forall \mathbf{x}$ Thus in order to satisfy the condition of Equation 6 λ_1 must be ≥ 0 and thus all such corresponding λ values shoud be ≥ 0

b. Prove that if the eingenvalues of A are positive then A is positive definite

Let us consider an arbitrary vector in the column space of the matrix A. Now we know that each vector in the column space can be represented as a linear combination of the corresponding eigen vectors. Now considering the condition for positive definite matrices mentioned in 6

$$\mathbf{x}^T A\mathbf{x} \geq 0 \tag{8}$$

can be written as

$$(a_1\mathbf{e}_1 + a_2\mathbf{e}_2 + a_3\mathbf{e}_3 + ...)^T \, A \, (a_1\mathbf{e}_1 + a_2\mathbf{e}_2 + a_3\mathbf{e}_3 + ...) \geq 0 \tag{9}$$

which can be written in the form of

$$\left(a_1^2\lambda_1 {\mathbf{e}_1}^T\mathbf{e}_1 + a_2^2\lambda_2 {\mathbf{e}_2}^T\mathbf{e}_2 + a_3^2\lambda_3 {\mathbf{e}_3}^T\mathbf{e}_3 + ...\right) \geq 0 \tag{10}$$

using the fact that Matrix Multiplication is Assosciative and Distributive and the eigen vectors of the symmetric matrices are orthogonal we derive Equation 10

from Equation 9. Thus for Equation 10 to be valid $\lambda_i \geq 0$ for all i in the space of eigen vectors.

c. The gradient vector will be given by :

$$\begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} = \begin{bmatrix} 2x \\ 4y \end{bmatrix}$$

d. The critical point will be given where gradient of f = 0. The c.p. here will be :

$$\begin{bmatrix} 2x \\ 4y \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

given by x= 0, y =0.

e. The Hessian Matrix is given by:

$$\begin{bmatrix} \frac{\partial^2 f}{\partial^2 x} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial^2 y} \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 0 & 4 \end{bmatrix}$$

f. Since the Hessian matrix of the above function is a constant it's nature is same for all points covered in the function. Finding the eigen values using the characteristic polynomial given by :

$$\lambda^2 - 6\lambda + 8 = 0 \tag{11}$$

we get λ = 2,4. Thus we see that the Hessian matrix H is positive definite at all points since H is symmetric and both the eigen values of H are positive which makes the C.P (0,0) a local minima.

Q3. This task has been divided into the following parts the first part deals with data prepearation. In this part all the necessary packages are loaded and the data is imported into the specific format.The next parts are divided as per the sub questions:

Part 1 :

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import math
from bokeh.plotting import figure, show, Figure
from bokeh.models import ColumnDataSource, Label
from bokeh.io import output_notebook
output_notebook()
dataFrameTraining = pd.read_csv("./data/DWH_Training.csv")
dataFrameTesting = pd.read_csv("./data/DWH_Test.csv")
```

maleHeightWeightMap = { 'height': [],
'weight': []
}
femaleHeightWeightMap = { 'height': [],
'weight': []
}

a

def getClusterCentroid(dataFrameTraining):
positiveSamplesCentroid = [0.0,0.0] negativeSamplesCentroid = [0.0,0.0]
negativeSamplesCount = 0.0
for index,row in dataFrameTraining.iterrows():
if row['gender'] == -1: negativeSamplesCentroid[0]+=row['height(cm)']
negativeSamplesCentroid[1]+=row['weight(kg)'] negativeSamplesCount+=1
femaleHeightWeightMap['height'].append(row['height(cm)'])
femaleHeightWeightMap['weight'].append(row['weight(kg)'])
else: positiveSamplesCentroid[0]+=row['height(cm)']
positiveSamplesCentroid[1]+=row['weight(kg)']
maleHeightWeightMap['height'].append(row['height(cm)'])
maleHeightWeightMap['weight'].append(row['weight(kg)'])
negativeSamplesCentroid[0]/=negativeSamplesCount
negativeSamplesCentroid[1]/=negativeSamplesCount
positiveSamplesCentroid[0]/=(len(dataFrameTraining)-negativeSamplesCount)
positiveSamplesCentroid[1]/=(len(dataFrameTraining)-negativeSamplesCount)
return positiveSamplesCentroid,negativeSamplesCentroid

b

def getHyperPlaneCoordinates(w, b, dataFrameTraining):
height = [] weight = []
for index, row in dataFrameTraining.iterrows():
x = row['height(cm)'] y = (-(w[0]/w[1])*x)-(b/w[1]) height.append(x)
weight.append(y)
return height,weight
def classify(w, b, dataFrameTraining):
classificationResult = []
for index, row in dataFrameTraining.iterrows():
result = ((w[0]*row['height(cm)'] + w[1]*row['weight(kg)'])+b) if result ¿ 0:
classificationResult.append(1) else: classificationResult.append(-1)
return classificationResult
positiveCentroid,negativeCentroid = getClusterCentroid(dataFrameTraining)
w = [0,0] w[0] = (2*(positiveCentroid[0] - negativeCentroid[0])) w[1] =
(2*(positiveCentroid[1] - negativeCentroid[1]))
b = (math.pow(negativeCentroid[0],2) + math.pow(negativeCentroid[1],2)) -
(math.pow(positiveCentroid[0],2) + math.pow(positiveCentroid[1],2))
height,weight = getHyperPlaneCoordinates(w,b,dataFrameTraining)
classificationResult = classify(w, b, dataFrameTraining)

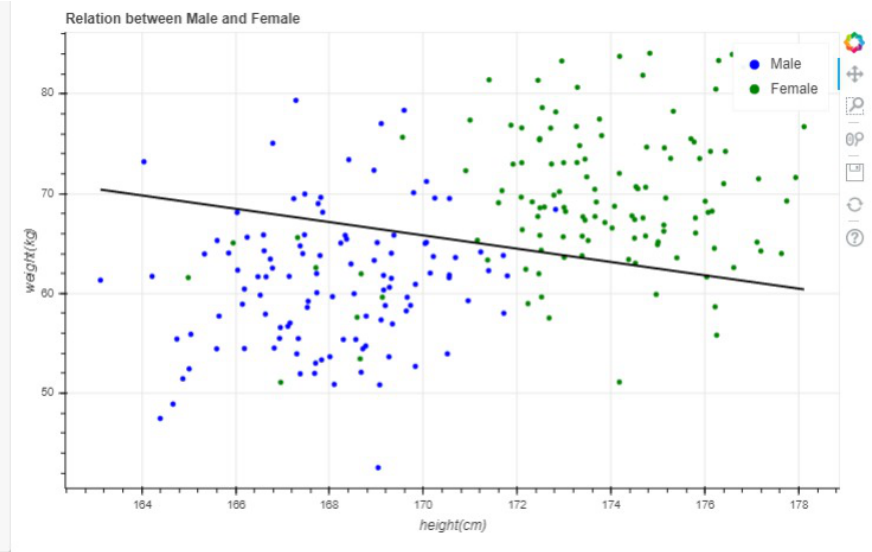

Figure 1: Scatter Plot Chart with the classifier line

c.

p = figure(x_axis_label='height(cm)',y_axis_label='weight(kg)', width=800, height=500,title='Relation between Male and Female') source = ColumnDataSource(data=maleHeightWeightMap) p.circle(x='height',y='weight',source=source,color="blue", legend="Male") source = ColumnDataSource(data=femaleHeightWeightMap) p.circle(x='height',y='weight',color="green",source=source, legend="Female") p.line(x=height,y=weight,line_width=2, color="black") show(p)

d.

testResults = classify(w,b, dataFrameTesting) correctClassificationCount = 0 for i,row in dataFrameTesting.iterrows(): if testResults[i] == row['gender']: correctClassificationCount +=1 print("Test Accuracy(

Test Accuracy : 75.777777 %