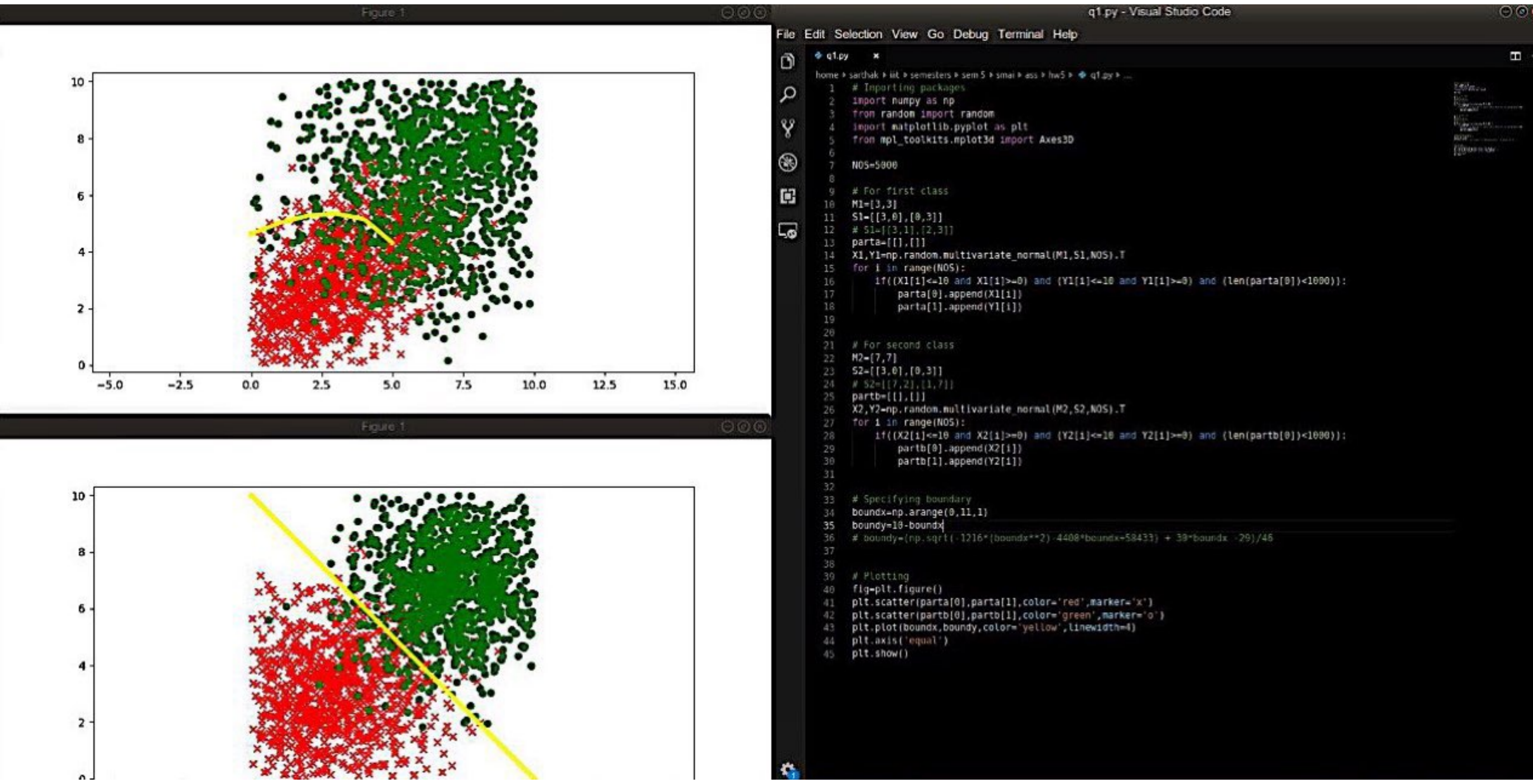
Figure 1
10
8
6
4
2
0
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0
12.5
15.0
Figure 1
q1.py - Visual Studio Code
File Edit Selection View Go Debug Terminal Help
q1.py
home ▸ sarthak ▸ iiit ▸ semesters ▸ sem 5 ▸ smai ▸ ass ▸ hw5 ▸ q1.py ▸ ...
# Importing packages
import numpy as np
from random import random
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
NOS=5000
# For first class
M1=[3,3]
S1=[[3,0],[0,3]]
# S1=[[3,1],[2,3]]
parta=[[],[]]
X1,Y1=np.random.multivariate_normal(M1,S1,NOS).T
for i in range(NOS):
if((X1[i]<=10 and X1[i]>=0) and (Y1[i]<=10 and Y1[i]>=0) and (len(parta[0])<1000)):
parta[0].append(X1[i])
parta[1].append(Y1[i])
# For second class
M2=[7,7]
S2=[[3,0],[0,3]]
# S2=[[7,2],[1,7]]
partb=[[],[]]
X2,Y2=np.random.multivariate_normal(M2,S2,NOS).T
for i in range(NOS):
if((X2[i]<=10 and X2[i]>=0) and (Y2[i]<=10 and Y2[i]>=0) and (len(partb[0])<1000)):
partb[0].append(X2[i])
partb[1].append(Y2[i])
# Specifying boundary
boundx=np.arange(0,11,1)
boundy=10-boundx
# boundy=(np.sqrt(-1216*(boundx**2)-4408*boundx+58433) + 39*boundx -29)/46
# Plotting
fig=plt.figure()
plt.scatter(parta[0],parta[1],color='red',marker='x')
plt.scatter(partb[0],partb[1],color='green',marker='o')
plt.plot(boundx,boundy,color='yellow',linewidth=4)
plt.axis('equal')
plt.show()

classmate
Date
Page

Ans1 Equation of decision boundary is given by the eq n:
$$[x-\mu_1]^T \Sigma^{-1} [x-\mu_1] = [x-\mu_2]^T \Sigma^{-1} [x-\mu_2]$$

1) $\Sigma_1 = \Sigma_2 = 3I$

$$\begin{bmatrix} x-3 \\ y-3 \end{bmatrix}^T \begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix}^{-1} \begin{bmatrix} x-3 \\ y-3 \end{bmatrix} = \begin{bmatrix} x-7 \\ y-7 \end{bmatrix}^T \begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix}^{-1} \begin{bmatrix} x-7 \\ y-7 \end{bmatrix}$$

$$\Rightarrow [x-3, y-3] \begin{bmatrix} x-3 \\ y-3 \end{bmatrix} = [x-7, y-7] \begin{bmatrix} x-7 \\ y-7 \end{bmatrix}$$

$$\Rightarrow (x-3)^2 + (y-3)^2 = (x-7)^2 + (y-7)^2$$

$$\Rightarrow 9-6x+9-6y = 49-14x+49-14y$$

$$\Rightarrow 18-6x-6y = 98-14x-14y$$

$$\Rightarrow 8x+8y = 80$$

$$\Rightarrow x+y = 10$$

$$\Rightarrow y = -x+10$$

2) $\Sigma_1 = \begin{bmatrix} 3 & 1 \\ 2 & 3 \end{bmatrix}$ $\Sigma_2 = \begin{bmatrix} 7 & 2 \\ 1 & 7 \end{bmatrix}$

$$\begin{bmatrix} x-3 \\ y-3 \end{bmatrix}^T \begin{bmatrix} 3 & 1 \\ 2 & 3 \end{bmatrix}^{-1} \begin{bmatrix} x-3 \\ y-3 \end{bmatrix} = \begin{bmatrix} x-7 \\ y-7 \end{bmatrix}^T \begin{bmatrix} 7 & 2 \\ 1 & 7 \end{bmatrix}^{-1} \begin{bmatrix} x-7 \\ y-7 \end{bmatrix}$$

$$\Rightarrow \frac{1}{7} [x-3, y-3] \begin{bmatrix} 3 & -1 \\ -2 & 3 \end{bmatrix} \begin{bmatrix} x-3 \\ y-3 \end{bmatrix} = LHS$$

$$\Rightarrow \cancel{LHS = \frac{1}{7}} \frac{1}{47} [x-7, y-7] \begin{bmatrix} 7 & -2 \\ -1 & 7 \end{bmatrix} \begin{bmatrix} x-7 \\ y-7 \end{bmatrix} = RHS$$

On solving we get:

$$y^* = [(-1216x^2 - 4408x + 58433)^{1/2} + 30x - 29]/46$$

When we plot these decision boundaries, we see that the boundaries seperate the classes.

Scanned with CamScanner

classmate

Date

Page

ans 2 a) Let student be a 200 x 1 vector & D be 2000 x 200 data matrix. According to the head of data analyst of zomato, there are only 35 different types of families that send their child to institute, based on their food preferences. So we can assume that all children belong to 35 classes only with slight variations & hence practical rank of D is 35. If definition of D_{ij} is changed, more categories may get introduced from each class & rank would increase. Thus rank could become anything from 35 - 100.

b) i) Break down D matrix into simpler matrices using SVD.

ii) Reconstruct D, by zeroing insignificant eigen values, to get a rank 35 D matrix.

iii) Now from the 1 month data, construct bit vector for each of the new 100 students.

iv) Apply KNN & find a neighbour which resembles student.

v) Assign that student to its neighbour's class.

vi) Using neighbour's bit vector, we would get a list of all those restaurants from which the new student might order.

vii) Now zomato could recommend the dishes from the restaurants of the given student.

c) This is similar to the previous one. Let us define a restaurant as d' x 1 vector where it has d' types of dishes. Using SVD, find practical rank of the matrix & say we have c categories of restaraunts. Based on transaction history, we can find how many students are likely to buy from a restaraunt & thus each restaraunt would be associated with certain students. Now apply KNN & find neighbour of new restaraunt. Then we can recommend this restaraunt to each of the student who is associated with the neighbour's restaraunt.

Ans 3 $Q = \frac{1}{N} \sum_{i=1}^{N} (x_i - \bar{x})(x_i - \bar{x})^T$

where $x_i = (x_{i1}, x_{i2} \ldots)^T$ & N = no. of samples.

For a nonzero $y \in R^k$:

$$y^T Q y = y^T \left(\frac{1}{N} \sum_{i=1}^{N} (x_i - \bar{x})(x_i - \bar{x})^T \right) y$$

$$= \frac{1}{N} \left(\sum_{i=1}^{N} y^T (x_i - \bar{x})(x_i - \bar{x})^T y \right)$$

$$= \frac{1}{N} \left(\sum_{i=1}^{N} \left((x_i - \bar{x})^T y \right)^T \left((x_i - \bar{x})^T y \right) \right)$$

$$= \frac{1}{N} \sum_{i=1}^{N} \left((x_i - \bar{x})^T y \right)^2 \geq 0$$

since $y^T Q y \geq 0$ & Q is a symmetric matrix, Q is a PSD.