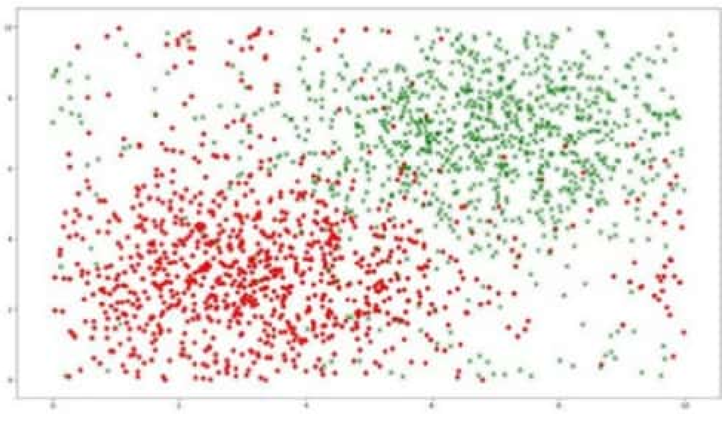
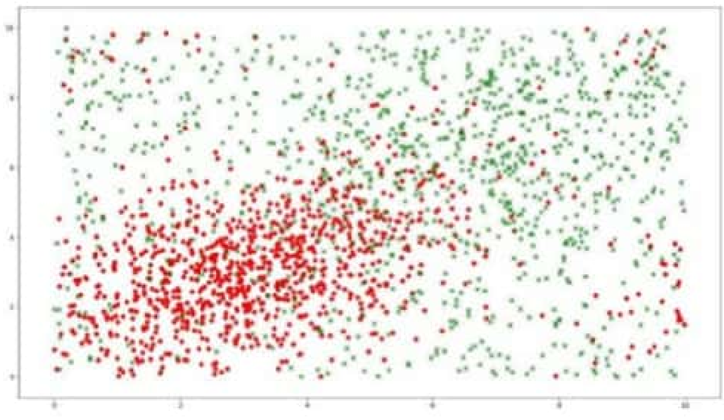
20171104

Q1) a) Using the numpy.random.multivariate_normal() function we create the plots by providing the given mean and covariance. To ensure that values are between [0, 10], I have taken mod 10.

```python
import numpy as np
import matplotlib.pyplot as plt
import scipy
from sklearn import svm

ele = 1000

mu1 = np.array([3, 3])
mu2 = np.array([7, 7])

sigma1 = np.array([[3, 0], [0, 3]])
sigma2 = np.array([[3, 0], [0, 3]])

# sigma1 = np.array([[3, 1], [2, 3]])
# sigma2 = np.array([[7, 2], [1, 7]])

cl1 = np.random.multivariate_normal(mu1, sigma1, ele) % 10
cl2 = np.random.multivariate_normal(mu2, sigma2, ele) % 10

fig = plt.figure()
plt.scatter(cl1[:, 0], cl1[:, 1], marker='o', color='red')
plt.scatter(cl2[:, 0], cl2[:, 1], marker='x', color='green')
plt.show()
```





Q1) b) i) $\mu_1 = [3, 3]^T$ $\mu_2 = [7, 7]^T$ $\Sigma_1 = \Sigma_2 = \Sigma = 3I$

$$\frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left(\frac{-1}{2}[x-\mu_1]^T \Sigma^{-1} [x-\mu_1]\right) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left(\frac{-1}{2}[x-\mu_2]^T \Sigma^{-1} [x-\mu_2]\right)$$

$$\Rightarrow [x-\mu_1]^T \Sigma^{-1} [x-\mu_1] = [x-\mu_2]^T \Sigma^{-1} [x-\mu_2]$$

$$\Rightarrow \left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - \begin{bmatrix} 3 \\ 3 \end{bmatrix}\right)^T \begin{bmatrix} 1/3 & 0 \\ 0 & 1/3 \end{bmatrix} \left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - \begin{bmatrix} 3 \\ 3 \end{bmatrix}\right) = \left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - \begin{bmatrix} 7 \\ 7 \end{bmatrix}\right)^T \begin{bmatrix} 1/3 & 0 \\ 0 & 1/3 \end{bmatrix} \left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - \begin{bmatrix} 7 \\ 7 \end{bmatrix}\right)$$

$$\Rightarrow \begin{bmatrix} x_1 - 3 & x_2 - 3 \end{bmatrix} \begin{bmatrix} 1/3 & 0 \\ 0 & 1/3 \end{bmatrix} \begin{bmatrix} x_1 - 3 \\ x_2 - 3 \end{bmatrix} = \begin{bmatrix} x_1 - 7 & x_2 - 7 \end{bmatrix} \begin{bmatrix} 1/3 & 0 \\ 0 & 1/3 \end{bmatrix} \begin{bmatrix} x_1 - 7 \\ x_2 - 7 \end{bmatrix}$$

$$\Rightarrow \frac{(x_1-3)^2}{3} + \frac{(x_2-3)^2}{3} = \frac{(x_1-7)^2}{3} + \frac{(x_2-7)^2}{3}$$

$$\Rightarrow x_1^2 - 6x_1 + 9 + x_2^2 - 6x_2 + 9 = x_1^2 - 14x_1 + 49 + x_2^2 - 14x_2 + 49$$

$$\Rightarrow x_1 + x_2 = 10$$

$\therefore$ Decision Boundary $\Rightarrow$ $x + y = 10$

Q1) b) ii) $\mu_1 = [3,3]^T$ $\mu_2 = [7,7]^T$ $\Sigma_1 = \begin{bmatrix} 3 & 1 \\ 2 & 3 \end{bmatrix}$ $\Sigma_2 = \begin{bmatrix} 7 & 2 \\ 1 & 7 \end{bmatrix}$

$|\Sigma_1| = 7$ $|\Sigma_2| = 47$

$\Sigma_1^{-1} = \begin{bmatrix} 3/7 & -1/7 \\ -2/7 & 3/7 \end{bmatrix}$ $\Sigma_2^{-1} = \begin{bmatrix} 7/47 & -2/47 \\ -1/47 & 7/47 \end{bmatrix}$

Using same approach as last question,

$$\frac{1}{\sqrt{7}} \exp\left(\frac{-1}{2}[x-\mu_1]^T \Sigma_1^{-1} [x-\mu_1]\right) = \frac{1}{\sqrt{47}} \exp\left(\frac{-1}{2}[x-\mu_2]^T \Sigma_2^{-1} [x-\mu_2]\right)$$

~~exp~~ Taking log on both sides,

$$\frac{1}{2}\left([x-\mu_1]^T \Sigma_1^{-1}[x-\mu_1] - [x-\mu_2]^T \Sigma_2^{-1}[x-\mu_2]\right) = \log\left(\frac{\sqrt{47}}{\sqrt{7}}\right)$$

$$\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - \begin{bmatrix} 3 \\ 3 \end{bmatrix}\right)^T \begin{bmatrix} 3/7 & -1/7 \\ -2/7 & 3/7 \end{bmatrix} \begin{bmatrix} x_1 - 3 \\ x_2 - 3 \end{bmatrix} - \begin{bmatrix} x_1 - 7 \\ x_2 - 7 \end{bmatrix}^T \begin{bmatrix} 7/47 & -2/47 \\ -1/47 & 7/47 \end{bmatrix} \begin{bmatrix} x_1 - 7 \\ x_2 - 7 \end{bmatrix} = \log\left(\frac{47}{7}\right)$$

$$92x_1^2 + 92x_2^2 - 120x_1x_2 + 116x_1 + 116x_2 = 2504 + 658 \log\sqrt{\frac{47}{7}}$$

$\therefore$ Decision Boundary $\Rightarrow$

$$92x^2 + 92y^2 - 120xy + 116x + 116y - \left(2504 + 329\log\frac{47}{7}\right) = 0$$

Q2) D contains binary representation of 2000 students & 200 restaurants

a) 1) Element D_{ij} is 1 if student i orders from restaurant j. Thus, D is a binary matrix. Since class labels of students can only be upto 35, thus matrix D has 35 independent features at the best (considering SVD).

Hence rank of matrix = 35

dimension = 2000×200

2) If D_{ij} contains number of times, then it represents the weights of preference. Hence, we are adding information about students and restaurant mapping.

Hence rank of matrix ≥ 35

b) Using SVD, a matrix is decomposed as-

$$D = U \Sigma V^T$$

Consider the matrix D containing binary information about 35 classes of students

$\dim(\Sigma) = \text{rank}(D) \times \text{rank}(D)$

$\dim(U) = 2000 \times \text{rank}(D)$

$\dim(V^T) = \text{rank}(D) \times 200$

Matrix U contains feature representation of students about choice of restaurant. [student → restaurant]

V contains feature representation of restaurants → students

Now, for 100 new students we have information about 5 orders. Thus, we can generate a sparse matrix D' for the new students using this information

$$D' = U' \Sigma' V^T$$

Σ' for D' will be taken with assumption that the new 100 students are not outliers. We reshape the matrix and find U' corresponding to nearest student weight in Σ. Using those values, recompute estimate of D'. Using some algorithm like Gradient Descent, we minimise error. Eventually, we find U' nearest to Σ and D.

U' is the recommendation system sent to the restaurant for the new students.

Algorithm:

1) Given $D \rightarrow$ perform SVD $\rightarrow$ obtain Σ_{ground}
2) Generate D' from 100 students and 5 orders
3) Perform SVD of D'. $Rank(\Sigma') < Rank(\Sigma_{ground})$
4) Use Σ_{ground} values and corresponding U values to form $D'' = U_{new} \Sigma_{ground} V^T$ such that we get $\min_{U_{new}} \| reshape(D) - D'' \|$
5) Return Unew as student information to restaurant

Scanned by CamScanner

c) For new restaurant, follow above algorithm for V.
Consider λ as small number of orders

Algorithm:

1) Generate Σ_{ground} from given D using SVD
2) Generate D' from new restaurant and orders
3) Recompute Σ' from D'
4) Take 35 values (assuming only 35 classes) from Σ' and find nearest corresponding $V^{T'}$ values such that the recomputed matrix D'' using $V^{T'}$ and Σ' follows $\min_{V^{T'}} \|reshape(D) - D''\|$
5) Return $V^{T'}$ to restaurant owner to send recommendation feature to students.

Q3) Prove that Covariance matrix is PSD.

For a sample of vectors, $x_i = (x_{i1}, x_{i2}, \dots, x_{ik})^T$ where $i = 1, 2, \dots n$,

Q = Covariance matrix = $\frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})(x_i - \bar{x})^T$ where $\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$

For a non-zero vector $y \in R^k$,

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

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

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

$$\Rightarrow y^T Q y \geq 0$$

$\Rightarrow$ Q or Covariance Matrix is PSD.