Ques 1-

$X = [x_1, x_2, \ldots, x_n] \quad x_i \in \mathbb{R}^d$

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \quad \begin{bmatrix} \text{mean vector for} \\ X \end{bmatrix}$$

$$\bar{x} = \left[\frac{x_1 + x_2 + x_3 + \ldots + x_n}{n}\right] \quad \bar{x} \in \mathbb{R}^d \quad \text{--- (1)}$$

(a.)

$$d_M^2(x_i, \bar{x}) = (x_i - \bar{x})^T (x_i - \bar{x})$$

distance b/w an in-sample point x_i & mean $\bar{x}$

$$x_i = \begin{bmatrix} a_1^i \\ a_2^i \\ \vdots \\ a_d^i \end{bmatrix} ; \quad \bar{x} = \frac{1}{n}\begin{bmatrix} a_1^1 + a_1^2 + \ldots a_1^n \\ a_2^1 + a_2^2 + \ldots a_2^n \\ \vdots \\ a_d^1 + a_d^2 + \ldots a_d^n \end{bmatrix} = \begin{bmatrix} \bar{a}_1 \\ \bar{a}_2 \\ \vdots \\ \bar{a}_d \end{bmatrix}_{d \times 1}$$

$$x_i - \bar{x} = \begin{bmatrix} a_1^i - \bar{a}_1 \\ a_2^i - \bar{a}_2 \\ \vdots \\ a_d^i - \bar{a}_d \end{bmatrix}_{d \times 1}$$

$$(x_i - \bar{x})^T = [a_1^i - \bar{a}_1 \quad a_2^i - \bar{a}_2 \quad \ldots \quad a_d^i - \bar{a}_d]_{1 \times d}$$

$$(x_i - \bar{x})^T (x_i - \bar{x}) = \begin{bmatrix} a_1^i - \bar{a}_1 & a_2^i - \bar{a}_2 & \cdots & a_d^i - \bar{a}_d \end{bmatrix} \begin{bmatrix} a_1^i - \bar{a}_1 \\ a_2^i - \bar{a}_2 \\ \vdots \\ a_d^i - \bar{a}_d \end{bmatrix}$$

$$= (a_1^i - \bar{a}_1) \cdot (a_1^i - \bar{a}_1) + (a_2^i - \bar{a}_2) \cdot (a_2^i - \bar{a}_2) \cdots\cdots + (a_d^i - \bar{a}_d) \cdot (a_d^i - \bar{a}_d)$$

OR

$$= x_i^T \cdot x_i + \bar{x}^T \cdot \bar{x} - x_i^T \cdot \bar{x} - \bar{x}^T \cdot x_i$$

$$= x_i^T \cdot x_i + \bar{x}^T \cdot \bar{x} - 2 x_i^T \cdot \bar{x} \quad \text{--- } (2)$$

(b)

$$K_{lin} = X^T X$$

$$= \begin{array}{c|cccc} & x_1 & x_2 & \cdots & x_n \\ \hline x_1 & x_1^T \cdot x_1 & x_1^T \cdot x_2 & \cdots & x_1^T \cdot x_n \\ x_2 & x_2^T \cdot x_1 & x_2^T \cdot x_2 & \cdots & x_2^T \cdot x_n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_n & x_n^T \cdot x_1 & x_n^T \cdot x_2 & \cdots & x_n^T \cdot x_n \end{array}$$

$$d_E^2(x_i, \bar{x}) = (x_i - \bar{x})^T (x_i - \bar{x})$$

$$= x_i^T x_i + \bar{x}^T \bar{x} - 2 x_i^T \bar{x} \quad (\text{from eq}^n \ (2))$$

$$\bar{x} = \left[\frac{x_1 + x_2 + x_3 + \cdots + x_n}{n}\right] \quad (\text{from eq}^n \text{ 1})$$

1 $x_i^T x_i = K_{lin}(i, i)$

2 $\bar{x}^T \bar{x} = \left(\frac{x_1 + x_2 + \cdots + x_n}{n}\right)^T \left(\frac{x_1 + x_2 + \cdots + x_n}{n}\right)$

$$= \frac{1}{n^2}\left[\sum_{i=1}^{n} x_i^T x_i + \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j \neq i}}^{n} x_i^T x_j\right] \quad \text{— 3}$$

(# Square of n vectors in the dot product form)

if x_i is 1D vector & $n = 2$

$$\therefore \left(\frac{x_1 + x_2}{2}\right)^T \left(\frac{x_1 + x_2}{2}\right) = \left(\frac{x_1 + x_2}{2}\right)^2$$

from eq^n 3

$$\frac{x_1^T x_1 + x_2^T x_2 + x_1^T x_2 + x_2^T x_1}{4} = \frac{x_1^2 + x_2^2 + 2x_1 x_2}{4}$$

$$\frac{x_1^2 + x_2^2 + 2x_1 x_2}{4} = x_1^2 + x_2^2 + 2x_1 x_2$$

$$= \frac{1}{n^2}\left[\sum_{i=1}^{n}\sum_{j=1}^{n} K_{lin}(i,j)\right]$$

(3) $$2x_i^T \bar{x} = 2\left(x_i^T\left(\frac{x_1 + x_2 + \dots x_n}{n}\right)\right)$$

$$= \frac{2}{n}\left[x_i^T x_1 + x_i^T x_2 + \dots\dots x_i^T x_n\right]$$

$$= \frac{2}{n}\left[\sum_{j=1}^{n} x_i^T x_j\right]$$

$$= \frac{2}{n}\left[\sum_{j=1}^{n} K_{lin}(i,j)\right]$$

Therefore $d_E^2(x_i, \bar{x})$ in term of K_{lin} is

$$d_E^2(x_i, \bar{x}) = K_{lin}(i,i) + \frac{1}{n^2}\left[\sum_{i=1}^{n}\sum_{j=1}^{n} K_{lin}(i,j)\right] - \frac{2}{n}\left[\sum_{j=1}^{n} K_{lin}(i,j)\right]$$

#(c) Now the mean $\bar{x}$ is changed to weighted mean $\tilde{x}$

$$\tilde{x} = \frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i=1}^{n} w_i} \qquad \{w_i \geq 0\}$$

$$\tilde{x} = \frac{w_1 x_1 + w_2 x_2 + \cdots w_n x_n}{(w_1 + w_2 + \cdots + w_n)}$$

$$K_{lin} = X^T X$$

$$= \begin{array}{c|c} & \begin{array}{cccc} x_1 & x_2 & \cdots & x \end{array} \\ \begin{array}{c} x_1 \\ x_2 \\ \vdots \\ x_n \end{array} & \begin{bmatrix} x_1^T \cdot x_1 & x_1^T \cdot x_2 & \cdots & x_1^T \cdot x_n \\ x_2^T \cdot x_1 & x_2^T \cdot x_2 & \cdots & x_2^T \cdot x_n \\ \vdots & \vdots & \ddots & \vdots \\ x_n^T \cdot x_1 & x_n^T \cdot x_2 & \cdots & x_n^T \cdot x_n \end{bmatrix} \end{array}$$

$$d_E^2(x_i, \tilde{x}) = (x_i - \tilde{x})^T (x_i - \tilde{x})$$

$$= x_i^T x_i + \tilde{x} \cdot \tilde{x} - 2 x_i^T \tilde{x}$$

① $x_i^T x_i = K_{lin}(i, i)$

②

$$\tilde{x}^T \tilde{x} = \left(\frac{w_1 x_1 + \dots w_n x_n}{w_1 + w_2 + \dots w_n}\right)^T \left(\frac{w_1 x_1 + \dots w_n x_n}{w_1 + w_2 + \dots w_n}\right)$$

$$= \frac{1}{W^2}\left[\sum_{i=1}^{n} w_i^2 x_i^T x_i + \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j \neq i}}^{n} (w_i \cdot w_j) x_i^T \cdot x_j\right]$$

$$\left\{ W = (w_1 + w_2 + w_3 + \dots w_n) \right\}$$

$$= \frac{1}{W^2}\left[\sum_{i=1}^{n} \sum_{j=1}^{n} (w_i \cdot w_j) K_{lin}(i, j)\right]$$

③ $2 x_i^T \tilde{x} = 2 x_i^T \left(\frac{w_1 x_1 + w_2 x_2 + \dots + w_n x_n}{w_1 + w_2 + \dots + w_n}\right)$

$$= 2\left[\frac{w_1 x_i^T x_1 + w_2 x_i^T x_2 + \dots w_n x_i^T x_n}{w_1 + w_2 + \dots + w_n}\right]$$

$$= \frac{2}{W}\left[\sum_{j=1}^{n} w_j \, x_i^T x_j\right]$$

$$= \frac{2}{W}\left[\sum_{j=1}^{n} w_j \, K_{lin}(i, j)\right]$$

Therefore $d_E^2(x_i, \tilde{x})$ in term of K_{lin} is

$$d_E^2(x_i, \tilde{x}) = K_{lin}(i, i) +$$

$$+ \frac{1}{W^2}\left[\sum_{i=1}^{n}\sum_{j=1}^{n} (w_i \cdot w_j) K_{lin}(i, j)\right]$$

$$- \frac{2}{W}\left[\sum_{j=1}^{n} w_j \, K_{lin}(i, j)\right]$$

(d.) INPUT ⊢ $X = [x_1, x_2, \dots, x_n]$ $x_i \in \mathbb{R}^d$

$$K_\phi(x_i, x_j) = \begin{bmatrix} \phi(x_1)^T \phi(x_1) & \cdots & \phi(x_1)^T \phi(x_n) \\ \vdots & \ddots & \\ \phi(x_n)^T \phi(x_1) & \cdots & \phi(x_n)^T \phi(x_n) \end{bmatrix}$$

(ϕ = Kernel funⁿ)

→ Pseudo K-mean algo.

→ for i in X:
  → for j in centroid:
    → (calculate distance b/w i & centroid j)
  → assign label based on minimum distance

→ Take the mean of the data points of mean cluster & repeat the above logic N times

→ To calculate the distance d_E^2 we will use K_{lin} as shown below

Case 1 :-

$$d_E^2(x_i, x_j) = (x_i - x_j)^T (x_i - x_j)$$

$$= x_i^T x_i + x_j^T x_j - 2x_i^T x_j$$

$$= K_\phi(i,i) + K_\phi(j,j) - 2K_\phi(i,j)$$

Case 2 :-

$$d_E^2(x_i, \bar{x}) = (x_i - \bar{x})^T (x_i - \bar{x})$$

$\boxed{\bar{x} = \text{mean of each cluster}}$

$$= K_\phi(i,i) + \frac{1}{n^2}\left[\sum_{i=1}^{n}\sum_{j=1}^{n} K_\phi(i,j)\right] - \frac{2}{n}\left[\sum_{j=1}^{n} K_\phi(i,j)\right]$$

(i, j only for the data pts. belonging to each cluster)

Case 1 :- Initialize the Centroids with random points from the given data & then calculate the distance using eqn given in "Case 1"

Case 2: after the 1st iteration we calculate the distance b/w the data point x_i & mean($\bar{x}$) of each cluster. To do this we use eqn from "Case 2"

→ Assigning labels is straight forward: assign the cluster label to the data point (x_i) which has the lowest distance.

Repeat the above steps N times.

N = no. of iterations

[For more details please look at code in "mid-sem.ipynb" file]

(e)

$$d_E^2(y, \tilde{x}) = ??$$

$y =$ out-sample point

distance b/w y & $\tilde{x}$ is

$$d_E^2(y, \tilde{x}) = y^T \cdot y + \tilde{x}^T \cdot \tilde{x} - 2y^T \tilde{x}$$

$\rightarrow$ $y^T \cdot y =$ Simple dot Product

$\rightarrow$ from 1-C we can write

$$\tilde{x}^T \cdot x = \frac{1}{W^2}\left[\sum_{i=1}^{n}\sum_{j=1}^{n}(w_i \cdot w_j) K_{lin}(i,j)\right]$$

$\rightarrow$ $2y^T\tilde{x} = 2y^T\left(\frac{w_1x_1 + w_2x_2 + \cdots + w_nx_n}{w_1 + w_2 + \cdots + w_n}\right)$

$$= 2\left[\frac{w_1 \cdot y^T \cdot x_1 + w_2 y^T x_2 + \cdots w_n y^T \cdot x_n}{w_1 + w_2 + \cdots + w_n}\right]$$

$$\left\{ W = \left(w_1 + w_2 + w_3 + \dots w_n \right) \right\}$$

$$= \frac{2}{W} \left[w_1 \cdot y^T \cdot x_1 + w_2 \, y^T x_2 + \dots w_n \, y^T \cdot x_n \right]$$

$$= \frac{2}{W} \left[\sum_{i=1}^{n} w_i \, y^T \cdot x_i \right]$$

Therefore :

$$d_E^2 (y, \tilde{x}) = y^T \cdot y +$$

$$+ \frac{1}{W^2} \left[\sum_{i=1}^{n} \sum_{j=1} (w_i \cdot w_j) \, K_{lin} (i, j) \right]$$

$$- \frac{2}{W} \left[\sum_{i=1}^{n} w_i \, y^T \cdot x_i \right]$$

→ Kernealized K-Means

everything would be same as in case of 1.(d). only thing that will be changed is the distance formula used

This time we will use the distance : $d_E^2(y, \tilde{x})$

AML-Course---MID-SEM

Submitted by: - Hemant Yadav (PhD20017) (hemantya@iiit.ac.in)

Question 2:-

ARI: Adjusted Rand Index
AMI: Adjusted Mutual Information
NMI: Normalized Mutual Information

+ **The code is present in the mid-sem.ipynb jupyter notebook. It is self explanatory. And the instructions to run it are in the README.md file.**
+ **The average scores are averaged over 10 runs**
+ **The dataset used as mentioned in the question.**
+ **PCA is used to reduce the number of dimensions such that the total energy retained is not less than 95%. Initially we had 748 dimension which after the PCA operation reduced to 121**
+ **1 and 2 K-means shown below are based on the kernel approach derived in question 1a. And 1b.**
+ **3 is simply the SK-learn based K-means implementation.**

Observations:

+ **The Gaussian kernel gives better results compared to the linear kernel.**
+ **The Gaussian kernel t-SNE plot has less overlap between the different clusters compared to the Euclidean one.**
+ **The reason is since the Gaussian kernel takes the basis to the infinite space and it's easy to make clusters in that space.**

The quantitative results are shown below:-

1. **Linear kernel K-means :-**
    a. Average score :
        i. ari:0.57
        ii. ami:0.61
        iii. nmi:0.61
    b. Best score :
        i. ari:0.64
        ii. ami:0.65
        iii. nmi:0.65

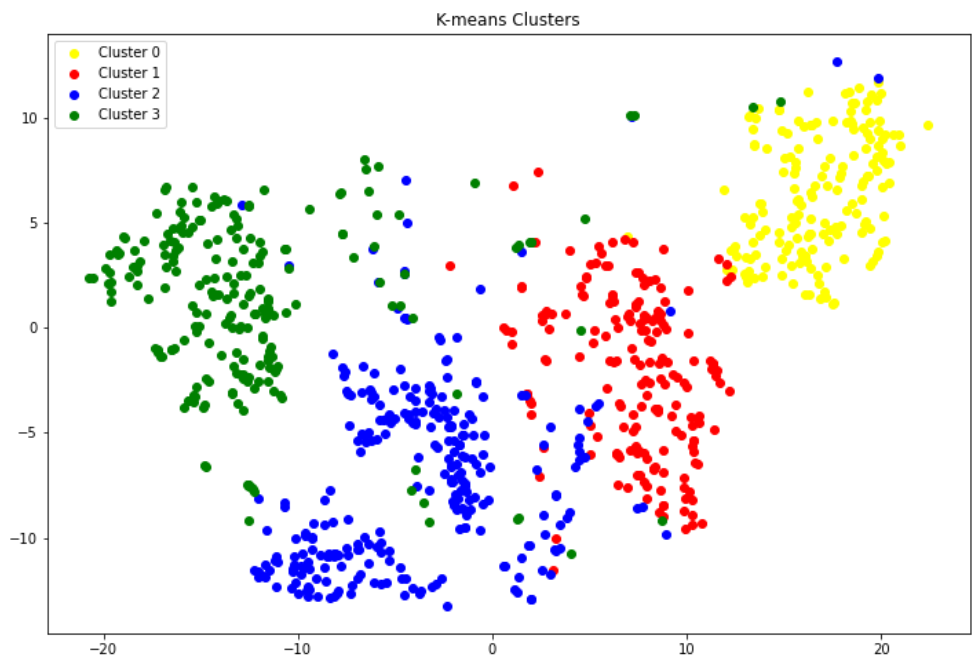

Figure 1. t-SNE plot of Linear Kernel implementation.

2. **Gaussian Kernel K-means:-** we ran the cross validation between (5 and 1e-10). The best value scale value was 1e-7

    a. Average score
        i. ari:0.59
        ii. ami:0.62
        iii. nmi:0.62
    b. Best score
        i. ari:0.73
        ii. ami:0.71
        iii. Nmi:0.71

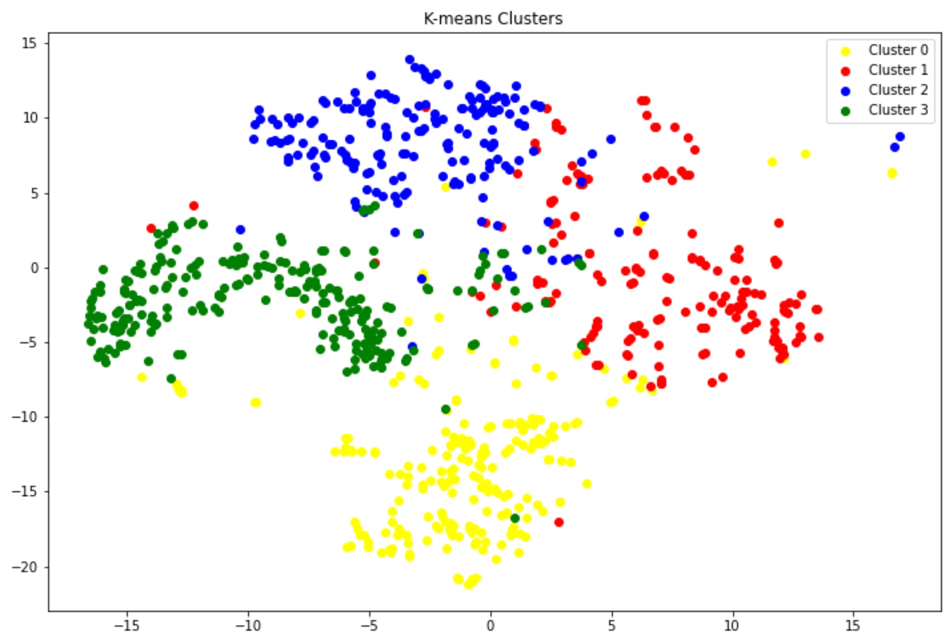

Figure 2. t-SNE plot of Gaussian Kernel implementation.

3. **Sk-learn usual K-means** (Euclidean distance calculation using points directly):-

   a. Average score:
      i. ari:0.47
      ii. ami:0.53
      iii. nmi:0.53
   b. Best score:
      i. ari:0.64
      ii. ami:0.65
      iii. Nmi:0.65

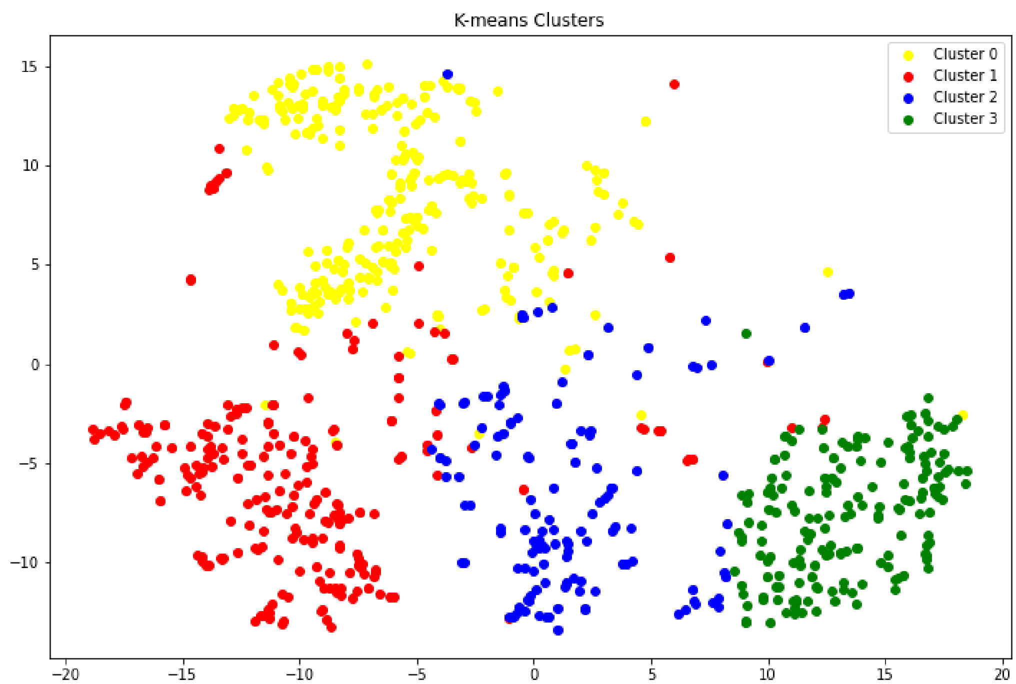

Figure 1. t-SNE plot of SK-learn implementation.