Some Studies on unsupervised multi-view clustering methods

Project Report

Prasun De
under
Dr. Swagatam Das
Associate Professor
Electronics and Communication Sciences Unit
Indian Statistical Institute
203 B T Road
Kolkata 700108

Indian Statistical Institute
India
August 11, 2024

Abstract

We study various approaches to unsupervised clustering of multi-dimensional data using both linear and non - linear algorithms .

Introduction

Clustering is a process which partitions a given data set into homogeneous groups based on given features such that similar objects are kept in a group whereas dissimilar objects are in different groups . It deals with finding structure in a collection of unlabeled data. As one of the classical clustering algorithms , **k-means** provides an intuitive and effective way to perform clustering. In specific, the k-means clustering is composed of (i) calculating k prototypes (i.e., centres of k clusters) given an assignment of samples to clusters and (ii) updating the assignment matrix by minimizing the sum-of-squares cost given the prototypes. These two steps are alternately performed until convergence. Due to its conceptual simplicity, easy-implementation and high efficiency, k-means clustering has been intensively studied and extended ([10] ; [9]; [11]; [12]). As an important extension, **kernel k-means** first maps data onto a high-dimensional space through a feature mapping and then conducts a standard k-means clustering in that space .In many practical applications of clustering, samples are represented by multiple groups of features extracted from different information sources. For example, three kinds of feature representations (colour, shape and texture) are extracted to distinguish one flower from another. These different sources usually provide complementary information, and it is better to let learning algorithms optimally combine them in order to obtain excellent clustering. This line of research is known as **multiple kernel clustering** in the literature.

1 k-means clustering

Let $\{x_i\}_{i=1}^n \subseteq \mathcal{X}$ be the set of n data-points .The objective of the **k-means** algorithm is to minimize the sum-of-squares loss over the *cluster assignment* matrix $\mathbf{Z} \in \{0,1\}^{n\times k}$. We can formulate it as the following optimization problem ,

$$\min_{Z\in\{0,1\}^{n,k}} \sum_{i=1,c=1}^{n,k} Z_{ic}|x_i-\mu_c|^2 \text{ such that } \sum_{c=1}^{k} Z_{ic}=1 \text{ where } \sum_{i=1}^{n} Z_{ic}=n_c, \text{ and } \mu_c=\frac{\sum_{i=1}^{n} Z_{ic}x_i}{n_c} \text{for } c=1,\cdots,k \quad (1)$$

We use the following algorithm to obtain **local optima** of the cost function . We have assumed *Lloyd's conditions* , that is , we will never encounter empty clusters during any stage of the procedure . Also note that k is either known , or we use G-means algorithm to obtain the most likely value of k. The k-means algorithm is highly sensitive to the initial (random) choice of the centroid matrix $\vec{\mu}=[\mu_1:\mu_2:\cdots\mu_k]$.

We shall describe the k means algorithm applied on the data-set X , with known number of clusters k .

Algorithm 1: kmeans(X , k)

Input: X , k
Output: Z , $\vec{\mu}$

1 Randomly initialize $\vec{\mu}=[\mu_1:\mu_2:\cdots\mu_k]$
2 Given $\vec{\mu}$ obtain Z by setting $Z_{ic}=1$ if c = argmin $\{\mathrm{d}(\mu_j,x_i)\}_{j=1}^{n}$ and 0 for other values of c (given i)
3 Given Z obtain $\vec{\mu}$ by setting $\mu_c=\frac{\sum_{i=1}^{n} Z_{ic}x_i}{n_c}$ where $n_c=\sum_{i=1}^{n} Z_{ic}$ for $c=1,\cdots,k$
4 Repeat Steps 2 and 3 until convergence of $\vec{\mu}$

2 Learning the k in k-means

When clustering a dataset, the right number k of clusters to use is often not obvious, and choosing k automatically is a difficult problem. The **G-means** algorithm [2] is an effective method to tackle this issue .

G-means repeatedly makes decisions based on a statistical test for the data assigned to each center. If the data currently assigned to a k-means center appear to be Gaussian, then we represent that data with only one center. However, if the same data do not appear to be Gaussian, then we use multiple centers to model the data properly. The algorithm will run

k-means multiple times (up to k times when finding k centers), so the time complexity is at most *O(k)* times that of k-means.

In the following algorithm : X denotes the data matrix , α is the level of significance of the statistical test . Widely used statistical tests for this purpose are the *Anderson - Darling test* and *Kolmogorov - Smirnoff test*

Algorithm 2: Gmeans(X , α)

Input: X , α
Output: k

1 Let C be initial set of centers .(Usually $\{C \leftarrow \bar{x}\}$)
2 C $\leftarrow$ k-means(C,X)
3 Let $\{x_i$—class$(x_i) = j\}$ be the set of datapoints assigned to center j .
4 Use statistical test to detect if each $\{x_i$—class$(x_i) = j\}$ follow a Gaussian distribution at confidence level α
5 If the data look Gaussian, keep c_j . Otherwise replace c_j with two centers.
6 Repeat from step 2 until no more centers are added.

For the **Anderson-Darling test** , the data can be then tested for uniformity with a distance test [4] . The formula for the test statistic A to assess if data $\{Y_1 < \cdots < Y_n\}$ (note that the data must be put in order) comes from a CDF F is

$$A^2 = -n - S\,,$$

where

$$S = \sum_{i=1}^{n} \frac{2i-1}{n} \left[\ln(F(Y_i)) + \ln\left(1 - F(Y_{n+1-i})\right)\right].$$

The test statistic can then be compared against the critical values of the theoretical distribution. Note that in this case no parameters are estimated in relation to the cumulative distribution function F .

3 Kernel K-means Clustering (KKM)

Let $\{x_i\}_{i=1}^{n} \subseteq \mathcal{X}$ be the set of n data-points , and $\phi(\cdot) : x \in \mathcal{X} \to \mathcal{H}$ be a feature mapping which maps x into a reproducing Kernel Hilbert Space $\mathcal{H}$.

The objective of the **k-means** algorithm is to minimize the sum-of-squares loss over the *cluster assignment* matrix $\mathbf{Z} \in \{0,1\}^{n\times k}$. We can formulate it as the following optimization problem ,

$$\min_{Z\in\{0,1\}^{n,k}} \sum_{i=1,c=1}^{n,k} Z_{ic}||\phi(x_i)-\mu_c||^2 \text{ such that } \sum_{c=1}^{k} Z_{ic} = 1 \text{ where } \sum_{i=1}^{n} Z_{ic} = n_c, \text{ and } \mu_c = \frac{\sum_{i=1}^{n} Z_{ic}\phi(x_i)}{n_c} \text{ for } c = 1,\cdots,k \tag{2}$$

We will now deduce an equivalent optimization problem from this . Let us define the **kernel matrix K** by $K_{ij} = \phi(x_i)^t\phi(x_j)$ for i , $j \in \{1,\cdots,n\}$. Now , we observe that ,

$$\begin{aligned}
&\sum_{\boldsymbol{i=1,c=1}}^{\boldsymbol{n,k}} \boldsymbol{Z_{ic}||\phi(x_i)-\mu_c||^2} \\
&= \sum_{\boldsymbol{i=1,c=1}}^{\boldsymbol{n,k}} \boldsymbol{Z_{ic}\left(\phi(x_i)-\mu_c\right)^T\left(\phi(x_i)-\mu_c\right)} \\
&= \sum_{\boldsymbol{i=1,c=1}}^{\boldsymbol{n,k}} \boldsymbol{Z_{ic}(\phi(x_i)^T-\mu_c^T)(\phi(x_i)-\mu_c)} \\
&= \sum_{\boldsymbol{i=1,c=1}}^{\boldsymbol{n,k}} \boldsymbol{Z_{ic}(\phi(x_i)^T\phi(x_i)-\phi(x_i)^T\mu_c-\mu_c^T\phi(x_i)+\mu_c^T\mu_c)} \\
&= \sum_{\boldsymbol{i=1}}^{\boldsymbol{n}}(\sum_{\boldsymbol{c=1}}^{\boldsymbol{k}} \boldsymbol{Z_{ic}})\boldsymbol{\phi(x_i)^T\phi(x_i)} - \sum_{\boldsymbol{i=1,c=1}}^{\boldsymbol{n,k}} \boldsymbol{Z_{ic}(\phi(x_i)^T\mu_c+\mu_c^T\phi(x_i)-\mu_c^T\mu_c)} \\
&= \sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} \boldsymbol{K_{ii}} - \sum_{\boldsymbol{i=1,c=1}}^{\boldsymbol{n,k}} \boldsymbol{Z_{ic}(\phi(x_i)^T\mu_c+\mu_c^T\phi(x_i)-\mu_c^T\mu_c)} \\
&= \sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} \boldsymbol{K_{ii}} - \sum_{\boldsymbol{c=1}}^{\boldsymbol{k}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} \boldsymbol{Z_{ic}(\phi(x_i)^T\mu_c+\mu_c^T\phi(x_i)-\mu_c^T\mu_c)}
\end{aligned}$$

Now , observe the second term in the summation on the previous line . We shall first simplify the inner-most summation that runs over $\boldsymbol{i}$. For this , we first observe a technical result :

If $\boldsymbol{A}$ is a $\boldsymbol{k} \times \boldsymbol{n}$ matrix , and $\boldsymbol{B}$ is a $\boldsymbol{n} \times \boldsymbol{n}$ matrix then :
$\left(\boldsymbol{ABA^T}\right)_{\boldsymbol{ij}} = \sum_{\boldsymbol{s=1}}^{\boldsymbol{n}} \sum_{\boldsymbol{t=1}}^{\boldsymbol{n}} \boldsymbol{a_{is} b_{st} a_{jt}}$

This can be seen easily by just expanding the LHS of the claimed equality . Now , we use this fact to proceed in our calculations :

$$\sum_{i=1}^{n} \boldsymbol{Z_{ic}} \left(\boldsymbol{\phi(x_i)^T \mu_c} + \boldsymbol{\mu_c^T \phi(x_i)} - \boldsymbol{\mu_c^T \mu_c}\right)$$

$$= \sum_{i=1}^{n} \boldsymbol{Z_{ic}} \left(\boldsymbol{\phi(x_i)^T \mu_c} + \boldsymbol{\mu_c^T \phi(x_i)} - \boldsymbol{\mu_c^T \mu_c}\right) - \frac{\left(\sum_{j=1}^{n} \boldsymbol{Z_{jc} \phi(x_j)^t}\right)\left(\sum_{j=1}^{n} \boldsymbol{Z_{jc} \phi(x_j)}\right)}{\boldsymbol{n_c^2}}$$

$$= \sum_{i=1}^{n} \boldsymbol{Z_{ic}} [\boldsymbol{\phi(x_i)^T} \cdot \frac{\sum_{j=1}^{n} \boldsymbol{Z_{jc} \phi(x_j)}}{\boldsymbol{n_c}} + \frac{\sum_{j=1}^{n} \boldsymbol{Z_{jc} \phi(x_j)}}{\boldsymbol{n_c}} \cdot \boldsymbol{\phi(x_i)}] - \frac{\left(\sum_{j=1}^{n} \boldsymbol{Z_{jc} \phi(x_j)^T}\right)\left(\sum_{j=1}^{n} \boldsymbol{Z_{jc} \phi(x_j)}\right)}{\boldsymbol{n_c^2}}$$

$$= \sum_{i=1}^{n} \boldsymbol{Z_{ic}} [\sum_{j=1}^{n} \frac{\boldsymbol{Z_{jc} K(x_i, x_j)}}{\boldsymbol{n_c}}] + \sum_{i=1}^{n} \boldsymbol{Z_{ic}} [\sum_{j=1}^{n} \frac{\boldsymbol{Z_{jc} K(x_i, x_j)}}{\boldsymbol{n_c}}] - \sum_{i=1}^{n} \boldsymbol{Z_{ic}} \sum_{j=1}^{n} \sum_{s=1}^{n} \frac{\boldsymbol{Z_{sc} Z_{jc} K(x_j, x_s)}}{\boldsymbol{n_c^2}}$$

$$= \boldsymbol{2} \cdot \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{\boldsymbol{Z_{ic} Z_{jc} K(x_i, x_j)}}{\boldsymbol{n_c}} - \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{s=1}^{n} \frac{\boldsymbol{Z_{ic} Z_{jc} Z_{sc} K(x_j, x_s)}}{\boldsymbol{n_c^2}}$$

$$= \boldsymbol{2} \cdot \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{\boldsymbol{Z_{ic} Z_{jc} K(x_i, x_j)}}{\boldsymbol{n_c}} - \sum_{j=1}^{n} \sum_{s=1}^{n} \sum_{i=1}^{n} \frac{\boldsymbol{Z_{ic}}}{\boldsymbol{n_c^2}} \cdot \boldsymbol{Z_{jc} Z_{sc} K(x_j, x_s)}$$

$$= \sum_{i=1}^{n}\sum_{j=1}^{n} \frac{\boldsymbol{Z_{ic}Z_{jc}K(x_i,x_j)}}{\boldsymbol{n_c}}$$

Now , observe that :

$$\sum_{i=1}^{n}\sum_{j=1}^{n} \frac{\boldsymbol{Z_{ic}Z_{jc}K(x_i,x_j)}}{\boldsymbol{n_c}} = \sum_{i=1}^{n}\sum_{j=1}^{n} \frac{\boldsymbol{Z_{ic}}}{\sqrt{\boldsymbol{n_c}}} \cdot \boldsymbol{K(x_i,x_j)} \cdot \frac{\boldsymbol{Z_{ic}}}{\sqrt{\boldsymbol{n_c}}}$$
$$= \boldsymbol{\left(AKA^T\right)_{cc}}$$

Here , $\boldsymbol{A}$ is a $\boldsymbol{k \times n}$ matrix such that : $\boldsymbol{A_{ij}} = \frac{\boldsymbol{Z_{ji}}}{\sqrt{\boldsymbol{n_i}}}$. Here , we have used the technical result that we had mentioned earlier . So , we get that :

$$\sum_{i=1,c=1}^{n,k} \boldsymbol{Z_{ic}||\phi(x_i) - \mu_c||^2 = Tr(K) - \sum_{c=1}^{k}\left(AKA^T\right)} \qquad (*)$$

Now , let $\boldsymbol{L} = \begin{bmatrix} \frac{1}{n_1} & 0 & 0 & \dots & 0 \\ 0 & 0 & \frac{1}{n_2} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \frac{1}{n_k} \end{bmatrix}$ Then , we observe that :

$$\boldsymbol{ZL^{1/2}} = \begin{bmatrix} z_{11} & z_{12} & z_{13} & \dots & z_{1k} \\ z_{21} & z_{22} & z_{23} & \dots & z_{2k} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ z_{n1} & x_{n2} & x_{n3} & \dots & x_{nk} \end{bmatrix} \times \begin{bmatrix} \frac{1}{\sqrt{n_1}} & 0 & 0 & \dots & 0 \\ 0 & \frac{1}{\sqrt{n_2}} & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \frac{1}{\sqrt{n_k}} \end{bmatrix}$$
$$= \begin{bmatrix} \frac{z_{11}}{\sqrt{n_1}} & \frac{z_{12}}{\sqrt{n_2}} & \frac{z_{13}}{\sqrt{n_3}} & \dots & \frac{z_{1k}}{\sqrt{n_k}} \\ \frac{z_{21}}{\sqrt{n_1}} & \frac{z_{22}}{\sqrt{n_2}} & \frac{z_{23}}{\sqrt{n_3}} & \dots & \frac{z_{2k}}{\sqrt{n_k}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{z_{n1}}{\sqrt{n_1}} & \frac{z_{n2}}{\sqrt{n_2}} & \frac{z_{n3}}{\sqrt{n_3}} & \dots & \frac{z_{nk}}{\sqrt{n_k}} \end{bmatrix}$$

So , we can take $\boldsymbol{A} = \left(\boldsymbol{ZL^{1/2}}\right)^{\boldsymbol{T}}$. Since $\boldsymbol{L}$ is symmetric , $\boldsymbol{A} = \boldsymbol{L^{1/2}Z^T}$. Plugging this value of $\boldsymbol{A}$ in $*$, we find that the objective function is equivalent to :

$$\boldsymbol{Tr(K) - \sum_{c=1}^{k}\left(L^{1/2}Z^TKZL^{1/2}\right)_{cc} = Tr(K) - Tr(L^{1/2}Z^TKZL^{1/2})}$$

We can say that the minimization problem :

$$\min_{Z\in\{0,1\}^{n,k}} \sum_{i=1,c=1}^{n,k} \boldsymbol{Z_{ic}}||\boldsymbol{\phi}(\boldsymbol{x_i}) - \boldsymbol{\mu_c}||^{\mathbf{2}} \text{ such that } \sum_{c=1}^{k} \boldsymbol{Z_{ic}} = \mathbf{1} \text{ where } \sum_{i=1}^{n} \boldsymbol{Z_{ic}} = \boldsymbol{n_c}, \text{ and } \boldsymbol{\mu_c} = \frac{\sum_{i=1}^{n} \boldsymbol{Z_{ic}}\boldsymbol{\phi}(\boldsymbol{x_i})}{\boldsymbol{n_c}} \text{ for } \boldsymbol{c} = \mathbf{1}, \cdots, \boldsymbol{k}$$

is equivalent to :
The optimization problem can thus be rewritten as :

$$\min_{Z\in\{0,1\}^{n,k}} \mathrm{Tr}(\mathbf{K}) - \mathrm{Tr}(\mathbf{L^{1/2}Z^{T}KZL^{1/2}}) \quad \boldsymbol{s.t.}\ \mathbf{Z1_k} = \mathbf{1_n} \tag{3}$$

Here , $\mathbf{K}$ is the kernel matrix , $\mathbf{L} = \mathbf{diag}[\boldsymbol{n_1^{-1}}, \cdots, \boldsymbol{n_k^{-1}}]$, and $\mathbf{1}_l \in \mathbf{R^l}$ is a column vector with all entries $\mathbf{1}$. The variable $\mathbf{Z}$ is discrete, which makes the op- timization problem very difficult to solve. However, we can approximate this problem through relaxing Z to take arbitrary real values. Specifically, by defining $\mathbf{H} = \mathbf{ZL^{1/2}}$ and letting $\mathbf{H}$ take real values , we obtain a relaxed version of the above problem .

$$\min_{H\in R^{n\times k}} \mathrm{Tr}(\mathbf{K}(\mathbf{I_n} - \mathbf{HH^{T}}) \quad \boldsymbol{s.t.} \quad \mathbf{H^{T}H} = \mathbf{I_k} \tag{4}$$

where $\mathbf{I_k}$ is the $\boldsymbol{k}\times\boldsymbol{k}$ identity matrix . Noting that $\mathbf{Z^{T}Z} = \mathbf{L^{-1}}$, we have $\mathbf{L^{1/2}Z^{T}ZL^{1/2}} = \mathbf{I_k}$. This leads us to the orthogonality constraint on $\mathbf{H}$. Finally, one can **obtain the optimal H by taking the k eigenvectors that correspond to the k largest eigenvalues of K** . We shall show it with the help of the following machinery that was developed in [2] .

Lemma from [8]

Let $\mathbf{y} \in \mathbf{R^q}$, $\mathbf{B^T} \in \mathbf{R^{q \times p}}$, such that $\mathbf{1} \leq \boldsymbol{q} \leq \boldsymbol{p}$**.** Consider the orthogonal linear transformation $\mathbf{x} \mapsto \mathbf{y} = \mathbf{B^T x}$. Let $\mathbf{\Sigma_y} = \mathbf{B^T \Sigma B}$ be the variance - covariance matrix for $\mathbf{y}$. Then , $\mathbf{Tr(\Sigma_y)}$ is maximized by taking $\mathbf{B = A_q}$, $\mathbf{A_q}$ consists of the first $\boldsymbol{q}$ columns of $\mathbf{A}$. Note that $\mathbf{A}$ is an orthogonal matrix whose $\boldsymbol{k}$-th column is the $\boldsymbol{k}$-th eigenvector of $\mathbf{\Sigma}$.

Proof of Lemma :
Let $\boldsymbol{\beta_k}$ be the $\boldsymbol{k}$-th column of $\mathbf{B}$. Now , the columns of $\mathbf{A}$ form a basis for $\mathbf{R^p}$. So , there exists $\{\mathbf{c_{jk}}\}_{j,k=1,1}^{j,k=p,q}$ such that $\boldsymbol{\beta_k} = \sum_{\mathbf{j=1}}^{\mathbf{p}} \mathbf{c_{jk}} \boldsymbol{\alpha_j}$ for $\boldsymbol{k = 1, \cdots, q}$. So , we may say that $\mathbf{B = AC}$. So ,

$$\begin{aligned}
\mathbf{B^T \Sigma B} &= \mathbf{C^T A^T \Sigma A C} \\
&= \mathbf{C^T \Lambda C} \\
&= \sum_{\mathbf{j=1}}^{\mathbf{p}} \boldsymbol{\lambda_j} \mathbf{c_j c_j^T}
\end{aligned}$$

Here , $\mathbf{c_j}$ is the $\boldsymbol{j}$-th row of $\mathbf{C}$

$$\begin{aligned}
\mathbf{Tr(B^T \Sigma B)} &= \sum_{\mathbf{j=1}}^{\mathbf{p}} \boldsymbol{\lambda_j} \mathbf{Tr(c_j c_j^T)} \\
&= \sum_{\mathbf{j=1}}^{\mathbf{p}} \boldsymbol{\lambda_j} \mathbf{Tr(c_j^T c_j)} \\
&= \sum_{\mathbf{j=1}}^{\mathbf{p}} \boldsymbol{\lambda_j} \mathbf{c_j^T c_j} \\
&= \sum_{\mathbf{j=1}}^{\mathbf{p}} \sum_{\mathbf{k=1}}^{\mathbf{q}} \boldsymbol{\lambda_j} \mathbf{c_{jk}^2} \ \cdots (i)
\end{aligned}$$

Now , $\mathbf{C = A^T B}$, and the orthogonality of $\mathbf{A}$ with orthonormal property of the columns of $\mathbf{B}$ implies that $\mathbf{C^T C = B^T A A^T B = B^T B = I_q}$. So ,

$\sum_{\mathbf{j=1}}^{\mathbf{q}} \sum_{\mathbf{k=1}}^{\mathbf{q}} \mathbf{c_{jk}^2} = \mathbf{q} \quad \cdots (\boldsymbol{ii})$ and the columns of $\mathbf{C}$ are also orthonormal . $\mathbf{C}$ can be thought of as the first $\mathbf{q}$ columns of a $\mathbf{p} \times \mathbf{p}$ orthogonal matrix , say , $\mathbf{D}$. Rows of $\mathbf{D}$ are orthonormal , so satisfy that $\mathbf{d_j^T d_j = 1}$ for $\mathbf{j = 1, \cdots, p}$. As rows of $\mathbf{C}$ consist of first $\boldsymbol{q}$ elements of rows of $\mathbf{D}$, $\mathbf{c_j^T c_j \leq 1}$. So , $\sum_{\mathbf{k=1}}^{\mathbf{q}} \mathbf{c_{jk}^2} \leq \mathbf{1} \cdots (\boldsymbol{iii})$. Now , $\sum_{\boldsymbol{k=1}}^{\boldsymbol{q}} \boldsymbol{c_{jk}^2}$ is coefficient of $\boldsymbol{\lambda_j}$ in $(\boldsymbol{i})$. So , the sum of these coeff. is $\mathbf{q}$ from $(\boldsymbol{ii})$. None of these can exceed $\mathbf{q}$ from $(\boldsymbol{iii})$. Now , since $\boldsymbol{\lambda_1 \geq \cdots \geq \lambda_p}$, it is clear that $\sum_{\mathbf{j=1}}^{\mathbf{p}} (\sum_{\mathbf{k=1}}^{\mathbf{q}} \mathbf{c_{jk}^2}) \boldsymbol{\lambda_j}$ will be maximized if we can find a set of $\mathbf{c_{jk}}$ for which :

$$\sum_{\mathbf{k=1}}^{\mathbf{q}} \mathbf{c_{jk}^2} = \begin{cases} \mathbf{1} & \mathbf{j = 1, \cdots, q} \\ \mathbf{0} & \mathbf{j = q+1, \cdots, p} \end{cases}$$

If $\mathbf{B^T = A_q^T}$, then

$$\mathbf{c_{jk}} = \begin{cases} \mathbf{1} & \mathbf{1 \leq j = k \leq q} \\ \mathbf{0} & \text{elsewhere} \end{cases}$$

So , $\mathbf{Tr(\Sigma_y)}$ achieves maximum when $\mathbf{B^T = A_q^T}$
This ends the proof of the lemma . ■

Now , we observe that solving $\min_{\boldsymbol{H \in R^{n \times k}}} \text{Tr}(\mathbf{K(I_n - HH^T)}$ $\boldsymbol{s.t.}$ $\mathbf{H^T H = I_k}$ is equivalent to solving
$\text{Tr}(\mathbf{K}) - \max_{\boldsymbol{H \in R^{n \times k}}} \text{Tr}(\mathbf{KHH^T})$ $\boldsymbol{s.t.}$ $\mathbf{H^T H = I_k}$. Now , by property of trace of a matrix , for two matrices $\mathbf{A, B}$ such that both $\mathbf{AB}$ and $\mathbf{BA}$ are defined , we have $\text{Tr}(\mathbf{AB}) = \text{Tr}(\mathbf{BA})$. As $\mathbf{K}$ is a fixed kernel matrix , it suffices to solve $\max_{\boldsymbol{H \in R^{n \times k}}} \text{Tr}(\mathbf{KHH^T})$ $\boldsymbol{s.t.}$ $\mathbf{H^T H = I_k}$. Taking $\mathbf{A = KH}$ and $\mathbf{B = H^T}$, we have to optimize
$\max_{\boldsymbol{H \in R^{n \times k}}} \text{Tr}(\mathbf{H^T KH})$ $\boldsymbol{s.t.}$ $\mathbf{H^T H = I_k}$.

Now , we shall make use of the lemma . Take $\mathbf{B = H}$, as $\mathbf{K}$ is the analogue of the variance- covariance matrix of the projected data-points in the feature space , it suffices to **obtain the optimal H by taking the k eigenvectors that correspond to the k largest eigenvalues of K** .

In the following , we shall discuss the algorithm for using **KKM** to obtain a clustering of the data-points .

Algorithm 3: kkmeans(X ,**K** , k)

Input: X ,**K** , k
Output: Z

1 Obtain the k eigenvectors that correspond to the k largest eigenvalues of $\mathbf{K}$: $\boldsymbol{v_1}, \cdots, \boldsymbol{v_k}$ and let $\boldsymbol{M} = [\boldsymbol{v_1} : \cdots, \boldsymbol{v_k}]$.
2 Normalize the rows of $\boldsymbol{M}$ and set $\mathbf{H}$ to be equal to this row-normalized $\boldsymbol{M}$.
3 Run **kmeans(H , k)** (Algorithm 2) and obtain Z .

4 Multiple Kernel K-means Clustering (MKKM)

In a multiple kernel clustering situation , as the name suggests , we have **multiple feature representations via a group of feature mappings** $\{\phi(.)\}_{p=1}^{m}$. Each sample can be represented as

$$\phi_{\mu}(\mathbf{x}) = [\mu_1\phi_1(\mathbf{x})^T, \cdots, \mu_m\phi_m(\mathbf{x})^T]^T.$$

, where $\boldsymbol{\mu} = [\mu_1, \cdots, \mu_m]^T$. denotes the coefficients of each base kernel that we need to optimize during the learning procedure . We can calculate the kernel function over the above mapping as follows :

$$\boldsymbol{K}_{\mu}(x_i, x_j) = \phi_{\mu}(x_i)^T\phi_{\mu}(x_j) = \sum_{p=1}^{m} \mu_p^2 \boldsymbol{K}_p(x_i, x_j).$$

We follow a procedure similar to what we did in the earlier case . Here , we are replacing $\boldsymbol{K}$ **with** $\boldsymbol{K_{\mu}}$**.** We obtain the objective function for the **MKKM** as follows :

$$\min_{H \in R^{n\times k}, \mu \in R_{+}^{m}} \mathrm{Tr}(\boldsymbol{K}_{\mu}(\mathbf{I_n} - \mathbf{HH^T})) \quad \boldsymbol{s.t.} \quad \mathbf{H^TH} = \mathbf{I_k}, \boldsymbol{\mu}^T\mathbf{1}_m = \mathbf{1}$$

We can solve this optimization problem as an **alternating minimization problem** by alternately updating $\boldsymbol{H}$ and $\boldsymbol{\mu}$**.**
(i) Optimizing H given $\boldsymbol{\mu}$. With $\boldsymbol{\mu}$ fixed , we can obtain $\boldsymbol{K_{\mu}}$ and then apply **KKM** (Algorithm 3) with input : { X , $\boldsymbol{K_{\mu}}$, k } . After the first

two steps of Algorithm 3 have been completed , we return the value of **H** . This is the updated value of **H** .

(ii) Optimizing μ given H . In this step , we have to solve the following quadratic programming problem that is subject to linear constraints.

$$\min_{\mu \in R_{+}^{m}} \sum_{p=1}^{m} \boldsymbol{\mu_p^2} \mathbf{Tr}(\mathbf{K_p}(\mathbf{I_n} - \mathbf{HH^T})) \;\; \boldsymbol{s.t.} \;\; \boldsymbol{\mu^T 1_m = 1}$$

We observe that this is a quadratic programming problem with non-negative constraints . This is a standard optimization problems that can be attacked with methods like gradient descent . Problems of this type have been discussed in [1] .

It has been seen that using a convex combination of the kernels , such as $\sum_{p=1}^{m} \boldsymbol{\mu_p K_p}$, often results in only kernel getting a nonzero weight and all others getting a zero weight . So , such a combination is not viable as it does not comply with the multi-view idea that is being deployed to ensure a better clustering of the data . Such situations have been discussed in [3] . Now , we shall write down the algorithm for using **MKKM** to obtain a clustering of the data points .

Algorithm 4: mkkmeans(X ,$\boldsymbol{\{K_p\}_{p=1}^{m}}$, k)

Input: X ,$\boldsymbol{\{K_p\}_{p=1}^{m}}$, k
Output: Z
1 Initialize $\boldsymbol{\mu^{(0)} = 1_m/m}$ and $\boldsymbol{t = 1}$.
2 Normalize the rows of $\boldsymbol{M}$ and set **H** to be equal to this row-normalized $\boldsymbol{M}$.
3 Run **kmeans(H , k)** (Algorithm 2) and obtain Z .

5 Multiple Kernel K Means with regularization

We should adequately consider the mutual influence of these kernels when updating kernel coefficients . So, the concept of *regularization* is introduced in [7] . To reduce the redundancy and enforce the diversity of the selected kernels, we need a regularization term that is able to characterise the correlation of each pair of kernels. To this end, we consider criterion $\boldsymbol{\mathcal{M}(K_p, K_q)}$

to measure the correlation between $\boldsymbol{K_p}$ and $\boldsymbol{K_p}$. A larger $\boldsymbol{\mathcal{M}(K_p, K_q)}$ means high correlation between $\boldsymbol{K_p}$ and $\boldsymbol{K_q}$, and a smaller one implies that their correlation is low. Therefore, a natural optimization criterion to prevent two highly correlated kernels from being selected can be defined as $\boldsymbol{\mu_p \mu_q \mathcal{M}(K_p, K_q)}$. As observed, by minimizing this term, the risk of simultaneously assigning $\boldsymbol{\mu_p}$ and $\boldsymbol{\mu_q}$ with large weights can be greatly reduced. Also, this regularization increases the probability of jointly assigning $\boldsymbol{\mu_p}$ and $\boldsymbol{\mu_q}$ with larger weights as long as $\boldsymbol{K_p}$ and $\boldsymbol{K_q}$ are less correlated. As a consequence, this criterion is beneficial to promote the diversity of selected kernels.

We have the following regularization term :

$$\min_{\mu \in R_+^m} \sum_{p,q=1}^{m} \boldsymbol{\mu_p \mu_q \mathcal{M}(K_p, K_q) = \mu^T M \mu \;\; s.t. \;\; \mu^T 1_m = 1} \tag{5}$$

By integrating the matrix-induced regularization into the objective function of existing **MKKM**, we obtain the optimization problem of the proposed algorithm as follows ,

$$\min_{\boldsymbol{H \in R^{n \times k}, \mu \in R_+^m}} \mathrm{Tr}(\boldsymbol{K_\mu}(\mathbf{I_n} - \mathbf{H}\mathbf{H}^\mathbf{T})) + \frac{\boldsymbol{\lambda}}{\mathbf{2}} \boldsymbol{\mu}^\mathbf{T} \mathbf{M} \boldsymbol{\mu} \quad \boldsymbol{s.t.} \quad \mathbf{H}^\mathbf{T}\mathbf{H} = \mathbf{I_k}, \boldsymbol{\mu}^T \mathbf{1}_m = \mathbf{1} \tag{6}$$

Here , $\boldsymbol{\lambda}$ is parameter to trade off the clustering cost function and the regularization term.

We have a two-step algorithm to solve the optimization problem in 6 alternatively.
(i) Optimizing **H** with fixed $\boldsymbol{\mu}$. Given $\boldsymbol{\mu}$, the optimization in 6 w.r.t **H** is a standard kernel k-means clustering problem, and the **H** can be obtained by solving 4 with given $\boldsymbol{K_\mu}$;
(ii) Optimizing $\boldsymbol{\mu}$ with fixed **H**. Given **H**, the optimization in 6 w.r.t $\boldsymbol{\mu}$ is a quadratic programming with linear constraints. In specific, we can obtain $\boldsymbol{\mu}$ by solving the following problem :

$$\min_{\mu \in R_+^m} \frac{\mathbf{1}}{\mathbf{2}} \boldsymbol{\mu^T (2Z + \lambda M) \mu} \;\; \boldsymbol{s.t.} \;\; \mathbf{H}^\mathbf{T}\mathbf{H} = \mathbf{I_k}, \boldsymbol{\mu}^T \mathbf{1}_m = \mathbf{1} \tag{7}$$

where $\boldsymbol{Z} = \text{diag}\left([\boldsymbol{Tr}(\boldsymbol{K_1}(\boldsymbol{I_n} - \boldsymbol{HH^T})), \cdots, \boldsymbol{Tr}(\boldsymbol{K_m}(\boldsymbol{I_n} - \boldsymbol{HH^T}))]\right)$.

Algorithm 5: mkkmeans-reg($\{\boldsymbol{K_p}\}_{\boldsymbol{p=1}}^{\boldsymbol{m}}, \boldsymbol{k}, \boldsymbol{\lambda}, \boldsymbol{\epsilon_0}$)

Input: $\{\boldsymbol{K_p}\}_{\boldsymbol{p=1}}^{\boldsymbol{m}}, \boldsymbol{k}, \boldsymbol{\lambda}, \boldsymbol{\epsilon_0}$
Output: $\boldsymbol{H}, \boldsymbol{\mu}$
Initialize $\boldsymbol{\mu^{(0)}} = \boldsymbol{1_m/m}$ and $\boldsymbol{t} = \boldsymbol{1}$.
repeat
  $\boldsymbol{K_\mu^{(t)}} = \sum_{\boldsymbol{p=1}}^{\boldsymbol{m}} \left(\boldsymbol{\mu_p^{(t-1)}}\right)^{\boldsymbol{2}} \boldsymbol{K_p}$
  Update $\boldsymbol{H^{(t)}}$ by solving **(4)** with given $\boldsymbol{K_\mu^t}$.
  Update $\boldsymbol{\mu^{(t)}}$ by solving **(7)** with given $\boldsymbol{H^t}$.
  t = t + 1
until $\left(\boldsymbol{obj^{(t-1)}} - \boldsymbol{obj^{(t)}}\right)/\boldsymbol{obj(t)} \leq \boldsymbol{\epsilon_0}$

We have to design $\boldsymbol{M}$ appropriate for a given clustering task. For example, $\boldsymbol{M(K_p, K_q)}$ could be defined according to some commonly used criteria such as Kullback-Leibler (KL) divergence [5] , maximum mean discrepancy (Gretton et al. 2006) [6] and Hilbert-Schmidt independence criteria (HSIC), to name just a few. One choice for this is $\boldsymbol{M(K_p, K_q)} = \boldsymbol{Tr(K_p^T K_q)}$. Designing proper $\boldsymbol{M}$ to satisfy various requirements of clustering tasks is interesting and worth exploring .

In this report , we have studied various unsupervised multi-view clustering methods that are used to group data points .

References

[1] Can Li A Conjugate Gradient Type Method for the Nonnegative Constraints Optimization Problems In *Journal of Applied Mathematics* Volume 2013, Article ID 986317

[2] Greg Hamerly and Charles Elkan. Learning the K in K-Means . In *Advances in Neural Information Processing Systems* vol 17 ,2004

[3] Bernhard Scholkopf , Alexander Smols , Klaus-Robert Muller. Nonlinear component analysis as a kernel eigenvalue problem. In *Neural Comput. 10(5):12991319.*

[4] Shapiro, S.S . How to test normality and other distributional assumptions In *The ASQC basic references in quality control: statistical techniques 3, pp. 178.*

[5] F. Topsoe . Some inequalities for information divergence and related measures of discrimination In *IEEE Transactions on Information Theory archive* Volume 46 Issue 4, July 2000 Pages 1602-1609

[6] A. Gretton , K.M. Borgwardt , M.J. Rasch, B. Scholkopf , A. Smola . A Kernel Two-Sample Test In *ournal of Machine Learning Research* Vol 13 (2012) pg 723-773

[7] Xinwang Liu, Yong Dou, Jianping Yin, Lei Wang, En Zhu Multiple Kernel k-Means Clustering with Matrix-Induced Regularization In *Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)*

[8] I.T. Jolliffe . Principal Component Analysis , 2nd ed. , Springer (2002)

[9] Gonen, M., and Margolin, A. A. . Localized data fusion for kernel k-means clustering with application to cancer biology. In *NIPS (2104) , 13051313.*

[10] Lu, Y.; Wang, L.; Lu, J.; Yang, J.; and Shen, C. 2014. Multiple kernel clustering based on centered kernel alignment. In *Pattern Recognition 47(11):3656 3664* (2014) .

[11] Cai, X.; Nie, F.; and Huang, H. 2013 . Multi-view k-means clustering on big data . In *IJCAI* .

[12] Du, L.; Zhou, P.; Shi, L.; Wang, H.; Fan, M.; Wang, W.; and Shen, Y.-D. 2015 . Robust multiple kernel k-means clustering using $\boldsymbol{l_{21}}$ norm. In *IJCAI , 34763482* .