On k-means Clustering for Complex High-Dimensional Data

GUAN XIN

MARCH 2025

On k-means Clustering for Complex High-Dimensional Data

A dissertation submitted to

THE GRADUATE SCHOOL OF ENGINEERING SCIENCE
OSAKA UNIVERSITY

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY IN SCIENCE

BY

GUAN XIN

MARCH 2025

Abstract

The k-means clustering is one of the most popular clustering methods, whereas it cannot perform well or even be inapplicable for data with non-linear cluster structure or with missing values, which is common in practice especially when the dimension of data is high.

Existing methods like kernel k-means and k-POD clustering have been proposed for such complex cases. However, both of them are ineffective for high-dimensional data, due to the existence of noise features that have no contribution to the underlying cluster structure. Therefore, the purpose of this thesis is to make the k-means-based clustering effective for high-dimensional data with non-linear cluster structure and missing values.

The first contribution is to propose the sparse kernel k-means clustering for high-dimensional data with non-linear cluster structure. It assigns each feature a binary indicator and conducts the kernel k-means clustering while penalizing the sum of the indicators. The proposed method extends the advantages of kernel k-means clustering that can capture the non-linear cluster structure to the high-dimensional cases. An alternative minimization algorithm is proposed to estimate both the cluster labels and the feature indicators. We also prove the consistency of both clustering and feature selection of the proposed method.

The second contribution is to propose the regularized k-POD clustering for high-dimensional missing data. It introduces a regularization function of cluster centers to k-POD clustering, which shrinks cluster centers feature-wisely. The proposed method can reduce the bias of estimated cluster centers and improve clustering performance, for the high-dimensional missing data, where noise features that have no contribution to cluster structure are common. In addition, we propose a general framework of optimization based on the majorization-minimization algorithm, which has convergence guarantee.

The experiments on synthetic datasets and applications on real-world datasets verify the effectiveness and show better performance of the proposed methods. As a consequence, we extend the application of traditional k-means clustering to more complex data in the big data age.

Contents

List of Figures

List of Tables

Notations

probability measure	$\mathbb{P}$
sample size	n
number of features (dimensions)	p
number of clusters	k
inner product of $x, \tilde{x} \in \mathbb{R}^p$	$\langle x, \tilde{x} \rangle_2 = \sum_{j=1}^p x_j \tilde{x}_j$
element-wise product of $x, \tilde{x} \in \mathbb{R}^p$	$x \circ \tilde{x} = (x_1 \tilde{x}_1, \ldots, x_p \tilde{x}_p)$
l_2 norm of $x \in \mathbb{R}^p$	$\|x\|_2 = \sqrt{\sum_{j=1}^p x_j^2}$
(In Chapter 3 and Appendix B)	$\|x\| = \sqrt{\sum_{j=1}^p x_j^2}$
l_1 norm of $x \in \mathbb{R}^p$	$\|x\|_1 = \sum_{j=1}^p \|x_j\|$
l_0 norm of $x \in \mathbb{R}^p$	$\|x\|_0 = \sum_{j=1}^p \mathbb{1}(x_j \neq 0)$
all-one vector in $\mathbb{R}^p$	$\mathbf{1}_p$
all-zero vector in $\mathbb{R}^p$	$\mathbf{0}_p$
indicator function	$\mathbb{1}(\cdot)$
identity matrix with size $n \times n$	I_n
the j-th column of matrix A	$\mathrm{A}_{(j)}$
the i-th row of matrix A	A_i
Frobenius norm of a matrix A	$\|\mathrm{A}\|_F = \sqrt{\sum_{i,j} a_{ij}^2}$
trace of an $n \times n$ matrix A	$\mathrm{tr}(\mathrm{A}) = \sum_{i=1}^n a_{ii}$
kernel function	$h(\cdot, \cdot)$
reproducing kernel Hilbert space	$\mathcal{H}$
inner product on $\mathcal{H}$	$\langle \cdot, \cdot \rangle_{\mathcal{H}}$
a norm on $\mathcal{H}$	$\|\cdot\|_{\mathcal{H}}$

Chapter 1

Introduction and preliminaries

This thesis investigates the challenges of clustering high-dimensional data with noise features, proposing novel k-means-based clustering methods to address non-linear cluster structure and missing values issues. The primary contribution lies in developing sparse kernel k-means clustering and regularized k-POD clustering, with applications in genomic data analysis.

1.1 Clustering

Clustering is one of the most classic tasks in the fields of Statistics and machine learning. It is a main technique of exploratory data analysis in Statistics, and also an important branch of unsupervised learning in machine learning. Compared with traditional regression and classification tasks, the most special point of clustering analysis is that we have no information about the response variable or labels for any data points.

The goal of clustering analysis is to group a set of unlabeled data points into several subsets, such that the data points belonging to the same group are more similar than those belonging to different groups. The obtained groups are called *clusters*, and the *clustering result* refers to the belonging relationship between data points and clusters. The clustering result depends on the measurement of similarity and the rule of belonging. First, the similarity (or dissimilarity) can be measured by different forms for different types of data and specific problems. The most common choice is to use the distance between two points to represent the dissimilarity between them, such as the Euclidean distance, Manhattan distance, as well as Minkowski distance and its generalizations for continuous data, Jaccard index for binary data and so on. The cosine similarity and some kernel functions are also widely-used to measure similarity. Second, the rule of belonging of data points to clusters can be deterministic or probabilistic, that is, each data point belongs to some unique group or belongs to any group with some probabilities, which is called *hard clustering* and *soft clustering*, respectively. With different similarity measurements and belonging rules, plenty of clustering methods have been proposed to characterize the latent cluster structures of data in various fields.

The methods of clustering can be based on connectivity between clusters, centroids of clusters, models, and density of data distribution. The most typical examples include hierarchical clustering, k-means clustering, Gaussian mixture model clustering, and DBSCAN. The hierarchical clustering gives a dendrogram, of which the leaf nodes are data points and the root node is a single cluster, and the stems show the path that these data points are merged to be clusters. Each hierarchy of the dendrogram gives a result of clustering. The k-means clustering gives k points to be cluster centers and assigns each data point to its nearest centers. The Gaussian mixture model clustering assumes that data points are independently sampled from a Gaussian mixture model and regards each component as a cluster. It gives the degree of each data point belonging to each cluster by the probability of it being generated from each component. The DBSCAN gives a result of clustering by gathering data points with their several nearest neighborhood points (in the sense that the distance is less than a threshold) and then identifies the rest isolated points to be noise points. Different clustering methods are suitable for different purposes of analysis and characteristics of data. In this thesis, we focus on the k-means clustering.

1.2 The k-means clustering

1.2.1 preliminaries

The k-means clustering is one of the most popular clustering methods. As a kind of centroids-based clustering, the core of the k-means clustering is the k cluster centers. The main idea is that those data points should belong to the same cluster if they are all similar to the same cluster center, where the dissimilarity is often measured by the Euclidean distance. Then, the k-means clustering groups data points by assigning each of them to its nearest cluster center.

Mathematically, suppose that data points $x_1, \ldots, x_n$ are independently drawn from the distribution $\mathbb{P}$, where we write $x_i = (x_{i1}, \ldots, x_{ip}) \in \mathbb{R}^p$ for any $i = 1, \ldots, n$. We also call the set $\{x_1, \ldots, x_n\}$ a sample. Let $\boldsymbol{\mu} = \{\mu_1, \ldots, \mu_k\}$ be the k cluster centers, where we write $\mu_l = (\mu_{l1}, \ldots, \mu_{lp}) \in \mathbb{R}^p$ for any

$l = 1, \dots, k$. Since the i-th data point x_i is assigned to the l^*-th cluster with

$$l^* = \mathop{\arg\min}_{l=1,\dots,k} \|x_i - \mu_l\|_2^2,$$

then the l-th cluster ($l = 1, \dots, k$) is given by

$$C_l = \{x_i, i = 1, \dots, n \mid \|x_i - \mu_l\|_2 \leq \|x_i - \mu_{l'}\|_2, \ \forall l' \neq l\} \tag{1.1}$$

Suppose that each data point is assigned to a unique cluster, then these C_l's are disjoint. We call that $\mathcal{C} = \{C_1, \dots, C_k\}$ is a *partition* of the set of data points.

In the k-means clustering, the optimal cluster centers are that minimize the sum of squares of distances between each data point and its nearest cluster center. Then, the objective (loss) function of k-means clustering is given by

$$\widehat{L}_n^{(\mathrm{KM})}(\boldsymbol{\mu}) = \sum_{i=1}^{n} \min_{l=1,\dots,k} \|x_i - \mu_l\|_2^2. \tag{1.2}$$

The estimated cluster centers $\hat{\boldsymbol{\mu}} = \{\hat{\mu}_1, \dots, \hat{\mu}_k\}$ are given by

$$\hat{\boldsymbol{\mu}} = \mathop{\arg\min}_{\boldsymbol{\mu}} \widehat{L}_n^{(\mathrm{KM})}(\boldsymbol{\mu}),$$

based on which, we can obtain the estimated partition $\widehat{\mathcal{C}} = \{\widehat{C}_1, \dots, \widehat{C}_k\}$ by Eq. (1.2). Since the minimization problem is known to be NP-hard, plenty of effective algorithms have been proposed to search for an approximate solution (local minima), such as Lloyd's algorithm and Hartigan–Wong method.

The Lloyd's algorithm first proposed by Lloyd (1982) is the the most classical one, which is also known as the standard algorithm for k-means clustering. It is a heuristic algorithm that alternatively updates the partition and cluster centers until convergence to the local minimum. Specifically, it considers the equivalent minimization problem with respect to cluster centers $\boldsymbol{\mu}$ and partition $\mathcal{C}$ as follows:

$$\min_{\boldsymbol{\mu}, \mathcal{C}} \sum_{l=1}^{k} \sum_{x_i \in C_l} \|x_i - \mu_l\|_2^2. \tag{1.3}$$

The algorithm starts with initialized cluster centers $\boldsymbol{\mu}^{(0)}$, and the $(t+1)$-th iteration ($t \in \mathbb{N}$) consists of two steps:

Step 1 Given cluster centers $\boldsymbol{\mu}^{(t)}$, update the partition by

$$C_l^{(t+1)} = \left\{x_i \mid \|x_i - \mu_l^{(t)}\|_2 \leq \|x_i - \mu_{l'}^{(t)}\|_2, \ \forall l' \neq l\right\}, \ \forall l = 1, \dots, k$$

Step 2 Given the partition $\mathcal{C}^{(t+1)}$, update cluster centers by

$$\mu_l^{(t+1)} = \frac{1}{|C_l^{(t+1)}|} \sum_{x_i \in C_l^{(t+1)}} x_i, \ \forall l = 1, \dots, k$$

The iteration procedure stops when the loss function Eq. (1.3) no longer changes under a small tolerance. Finally, the Lloyd's algorithm gives the current values of $\boldsymbol{\mu}$ and $\mathcal{C}$ as the estimated cluster centers and partition of k-means clustering. The convergence of the Lloyd's algorithm is guaranteed because essentially the two iterative steps are a kind of alternatively minimization, which makes the decreasing trend of objective function of Eq. (1.3) in each iteration.

1.2.2 Equivalent forms and relationship to other methods

Expect for the standard definition Eq. (1.2), the k-means clustering has several equivalent expressions, which suggest close relationship to other methods, especially principle component analysis (PCA) and matrix decomposition.

First, the k-means clustering can be viewed as a super sparse version of PCA (Zha et al. 2001, Ding & He 2004). Write X for the $n \times p$ data matrix consisting of $x_1, \dots, x_n$. Based on the fact that μ_l associated with C_l is given by the average of data points in C_l, the k-means clustering has another equivalent formulation:

$$\begin{aligned} &\min_{\mathrm{U}} \operatorname{tr}(\mathrm{XX}^T) - \operatorname{tr}(\mathrm{U}^T\mathrm{XX}^T\mathrm{U}) \\ &\text{s.t. } \mathrm{U} \in \{0,1\}^{n \times k}, \ \mathrm{U}\mathbf{1}_k = \mathbf{1}_n, \end{aligned} \tag{1.4}$$

where $\mathrm{U} = (u_{il})_{n \times k}$ and u_{il} represents whether x_i belongs to C_l. If we relax U to be arbitrary orthonormal matrix $\mathrm{U}^T\mathrm{U} = \mathrm{I}_k$, then this minimization is equivalent to the problem of PCA. We can thus regard the solution of PCA as a relaxed solution of k-means clustering. The Eq. (1.4) will be used in Chapter 2 of this thesis.

Second, the k-means clustering is a special case of matrix decomposition (Ding et al. 2005, Kim & Park 2008) and can be expressed as follows:

$$\begin{aligned} &\min_{\mathrm{U,M}} \|\mathrm{X} - \mathrm{UM}\|_F^2 \\ &\text{s.t. } \mathrm{U} \in \{0,1\}^{n \times k}, \ \mathrm{U}\mathbf{1}_k = \mathbf{1}_n, \end{aligned} \tag{1.5}$$

where $\mathrm{U} = (u_{il})_{n\times k}$ and u_{il} represents whether x_i belongs to C_l, and $\mathrm{M} = (\mu_{lj})_{k\times p} \in \mathbb{R}^{k\times p}$ is the codebook and the l-th row is μ_l. Furthermore, the k-means clustering is also viewed as a technology of *vector quantization*, which models the data distribution by several prototype vectors and is originally used for data compression. We will use the form Eq. (1.5) in Chapter 3.

In addition, the k-means clustering also has a close connection with other clustering methods like Gaussian mixture model clustering (Bishop 2006). Specifically, if all components of Gaussian mixture model have the same known covariance matrix $\sigma^2\mathrm{I}_p$, then the k-means using the Lloyd's algorithm and the Gaussian mixture model using EM algorithm have the equivalent objective functions. Moreover, when $\sigma^2 \to 0$, these two methods tend to give the same clustering results. The k-means clustering is thus viewed as the limiting case or hard version of Gaussian mixture model clustering.

1.2.3 The number of clusters

In the k-means clustering, the number of clusters k has a crucial influence on the clustering result. However, it is difficult to define the true number of clusters. Thus, we often select the best k from a set of candidates before conducting k-means clustering. The selection of k is often based on some heuristic criteria, such as information criterion (AIC and BIC), gap statistics, and instability.

The AIC and BIC are based on the likelihood function. Considering that the k-means clustering is a limiting version of Gaussian mixture model clustering, we can assume the data distribution is a mixture model with the same and known covariance and then calculate the value of AIC and BIC for each candidate k (Fraley & Raftery 2002, Hofmeyr 2020).

The gap statistics measures the difference between the within-cluster dispersion of the original data and expected that of null data that does not contain subgroups. The null data can be obtained by independently permuting observations of the original data matrix in each feature. We can then calculate the expected within-cluster dispersion of null data by averaging that of different null data (Tibshirani et al. 2002).

The instability measures how stable a clustering method is. It serves as the cross-validation in the field of clustering. It estimates cluster centers by conducting k-means clustering on two training datasets, and calculates the disagreement of prediction results of labels on a testing dataset. The training and testing datasets can generated by random splitting or bootstrap (Wang 2010, Fang & Wang 2012).

It should be noted that although these criteria were proposed originally for selecting the number of clusters k, they can also be used for tuning other parameters in clustering. In this thesis, we will apply the gap statistics in Chapter 2, and the BIC and instability in Chapter 3, where we will further introduce details of these criteria.

The k-means clustering has been widely applied in many fields from its first proposed by Steinhaus et al. (1956), such as computer science, biology, psychology, astronomy and business over the past 70 years due to its easy and fast implementation. Even in 2020s, the era of AI, the k-means clustering as well as its variants and adaptions still play an important role in addressing the complex analysis demands of big data. Among these demands of diverse data, the high-dimensionality of data particularly makes the k-means clustering challenging, and addressing the high-dimensional clustering is critical for modern applications such as genomics, where feature sparsity is common. Therefore, we focus on the high-dimensional data in this thesis.

1.3 The k-means clustering for high-dimensional data

In the big data age, it is very common for the data to be high-dimensional and with complex structure. A typical characteristic of high-dimensional data is the existence of noise dimensions, which we call "noise feature" in this thesis. *In the field of clustering, the noise feature is defined to be one that has no contribution to clustering.* For example, in genomic data, many genes do not contribute to the disease outcome, thus considered as noise features for the specific analysis.[1] The k-means clustering often works well for low-dimensional data and even with few noise features. However, due to the existence of plenty of noise features in high-dimensional data, traditional k-means clustering could fail to give a reasonable clustering result.

Example 1.1. *The Lymphoma dataset is a particular real-world case of high-dimensional data for clustering, which consists of 62 sample points ($n = 62$) collected from 3 types of cells of patients ($k = 3$). Out of the total 4026 gene expressions ($p = 4026$), more than 90% are noise features and have no contribution to clustering. If we apply the standard k-means clustering directly on the data with all features, the error rate of clustering is about 0.3. Figure 1.1 illustrates the estimated labels by the color of each dot, where the shape of each dot is the ground truth of labels. However, in this case, the*

[1] It should be noted that since the real "ground truth" about cluster labels of data points is unknown, we can not judge whether a feature is related or noise according to the correlation between it and the true label.

error rate on data with only relevant features is only 0.05. This suggests the importance of eliminating the negative influence of noise features for k-means clustering.

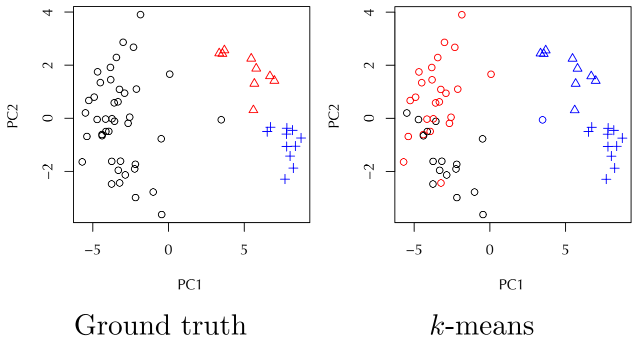


Figure 1.1: The result of k-means clustering on *Lymphoma* dataset. The x-axis and y-axis are the first two principle components of PCA conducted on original data points. The left panel is the ground truth and the right panel is the k-means clustering result, where the color of each dot represents the estimated label and the shape is the true label.

The methods to cope with high-dimensional data for k-means clustering mainly fall into two categories: feature selection and regularization.

Feature selection: The feature selection for clustering is tailored to choose a subset of features and use them for clustering data points. The main idea is to consider a non-negative weight for each feature of data and to optimize the weighted version of Eq. (1.2) with respect to cluster centers and feature weights, and in this way, the sparse solution of weights shows which features are selected and relevant to clustering.

Specifically, let $\omega = (\omega_1, \dots, \omega_p)$ be the weight vector for p features, where $\omega_j \geq 0$ for any $j = 1, \dots, p$. Then a weighted version of objective function is given by

$$\sum_{i=1}^{n} \min_{l=1,\dots,k} \sum_{j=1}^{p} \omega_j (x_{ij} - \mu_{lj})^2. \tag{1.6}$$

Different constraints on ω are used to get reasonable solutions. For example, Friedman & Meulman (2004) combine the constraint $\|\omega\|_1 = 1$ with $\sum_{j=1}^{p} \omega_j \log \omega_j \leq s$, whereas the solution is not sparse. The sparse solution of ω can be obtained by using the constraint $\|\omega\|_1 = 1$ and substituting ω_j of Eq. (1.6) by ω_j^β, where $\beta > 1$ is a pre-specified constant, as proposed by

Huang et al. (2005). Alternatively, Witten & Tibshirani (2010) use the constraints $\|\omega\|_2 = 1$ and $\|\omega\|_1 \leq s$, and they transform the objective function to an equivalent formulation based on the relationship of total variance, within-cluster variance and between-clusters variance, so that the trivial solution can also be avoided. We will use similar trick in Chapter 2.

In addition to using weights, one can also rank all p features by measuring the difference of total variance and within-cluster variance in each feature (Zeng & Cheung 2009, Zhang et al. 2020), or by testing the multi-modality of the density in each feature (Chan & Hall 2010, Jin & Wang 2016). However, the ranking result does not directly give a subset of relevant features.

Regularization: The regularization for clustering is aimed to yield cluster centers that have the same values in some features. The main idea is based on the following assumption: The j-th feature ($j = 1, \dots, p$) is called noise if the k cluster centers satisfy $\mu_{1j} = \cdots = \mu_{kj}$. Accordingly, adding a penalty on cluster centers to the original k-means objective helps to obtain such a solution.

Specifically, let $J : \mathbb{R}^k \to \mathbb{R}$ be a regularization function and $\{\lambda_j\}_{j=1}^p$ adaptive regularization parameters. Then the regularized version of Eq. (1.2) is given by

$$\sum_{i=1}^{n} \min_{l=1,\dots,k} \|x_i - \mu_l\|_2^2 + \sum_{j=1}^{p} \lambda_j J(\mu_{1j}, \dots, \mu_{kj}). \tag{1.7}$$

The penalty term is the summation of regularization on each feature. It follows that the large dispersion of cluster centers in a feature is more likely to be penalized. Write $\mathrm{M} \in \mathbb{R}^{k \times p}$, the l-th row of which is the l-th cluster center μ_l. Denote by $\mathrm{M}_{(j)}$ the j-th column. The common used forms of $J(\mathrm{M}_{(j)})$ for a sparse solution include $\|\mathrm{M}_{(j)}\|_1$ and $\|\mathrm{M}_{(j)}\|_0$ by Sun et al. (2012) and Raymaekers & Zamar (2022). We will use similar technique in Chapter 3.

It should be noted that, although feature selection methods for clustering have shown good performance in high-dimensional data according to numerical experiments, there are a few discussions on its asymptotic properties, such as the consistency of clustering and feature selection (Chang et al. 2018, Zhang et al. 2020). The difficulties mainly lie in the identifiability of “true cluster centers” and “noise features”, which rely on some considerably strong assumptions. On the other hand, regularization methods for clustering usually have good statistical guarantees (Sun et al. 2012, Levrard 2018, Raymaekers & Zamar 2022), which are based on some well-built theoretical frameworks for k-means clustering and under assumptions on specific forms of ground truth of cluster centers.

Furthermore, as the increasing demands for deep and precise analysis of real-world high-dimensional data, in practical applications, the k-means clustering for high-dimensional data also faces other complex issues, except for the existence of noise features. In this thesis, we mainly focus on two kinds of complex high-dimensional data: (1) The ground truth of cluster structure is non-linearly separable; (2) The data matrix has missing values. The methods of feature selection and regularization can help to recognize noise features, while unable to deal with non-linear cluster structure or missing values in high-dimensional data. Therefore, it is necessary to propose novel improvement of k-means clustering to address these problems, so as to satisfy the practical analysis demands for complex high-dimensional data in real-world applications. We will introduce the two kinds of complex high-dimensional data and the difficulties to deal with them in next section.

1.4 Two kinds of complex high-dimensional data

1.4.1 Non-linear cluster structure

Since the k-means clustering gives a partition in the form of Eq. (1.1), the clusters boundaries are linear in $\mathbb{R}^p$. Specifically, given l-th and l'-th cluster centers μ_l and $\mu_{l'}$, the boundary between two clusters is given by $f(x) = 0$, where

$$f(x) = \langle \mu_l - \mu_{l'}, x \rangle_2 - \frac{1}{2}\left(\|\mu_l\|_2^2 - \|\mu_{l'}\|_2^2\right)$$

is a linear function with respect to $x \in \mathbb{R}^p$. For any $x \in \mathbb{R}^p$ (not necessarily the sample points), whether it should be assigned to l-th cluster or l'-th cluster is decided by the sign of $f(x)$. Therefore, instead of grouping the sample, we can also use these linear boundaries to divide the whole data space into k convex hulls, which are also called *Voronoi cells*. In this thesis, when the ground truth of cluster structure (i.e., true partition of data space) can be characterized by linear boundaries, we call it by *linear cluster structure*. In addition, we also call the data with linear cluster structure as *linear data* for short. Figure 1.2 gives toy examples for different cluster structures in $\mathbb{R}^2$.

When the ground truth of cluster structure is non-linear, using k-means clustering to partition the sample and data space is obviously not appropriate. Moreover, in high-dimensional data, even if the noise features have been eliminated by using feature selection and regularization, the cluster structure

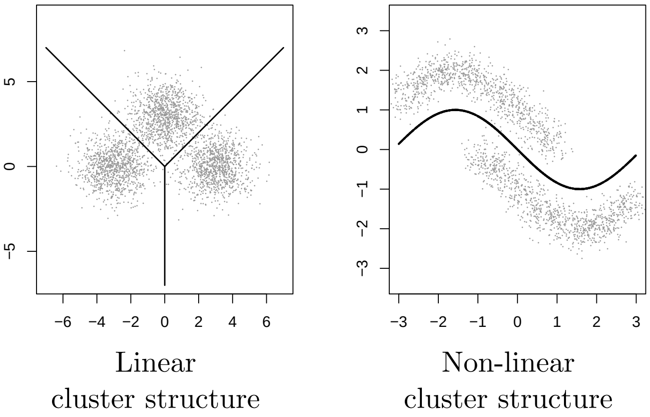


Figure 1.2: The examples of linear cluster structure and non-linear cluster structure in $\mathbb{R}^2$. The lines are boundaries between clusters.

in relevant features can be quite complex. It suggests that apart from coping with noise features, we need to handle non-linear cluster structure as well.

To do so, a popular method is to transform the data. That is, consider a mapping $\psi : \mathcal{X} \to \mathcal{H}$, where $\mathcal{X}$ is the data space in $\mathbb{R}^p$, such that the cluster structure of transformed data $\psi(x)$ is linear in the space $\mathcal{H}$. Since the mapping is usually connected to a kernel function (Fukumizu 2010), the transformed version is called *kernel k-means clustering* (Dhillon et al. 2004). We will introduce the detail in Chapter 2.

The kernel k-means clustering is famous for its applicability to characterize the non-linear cluster structure of data. However, since the mapping considers the data space of all p features, the existence of plenty of noise features also has a negative influence on the kernel k-means clustering, and even makes it fail. On the other hand, the existing feature selection method (Eq. (1.6)) is only for k-means clustering.

Example 1.2. *The ORL data is a face image dataset with 1024 pixel values ($p = 1024$), consisting of 40 images ($n = 40$) from 4 distinct subgroups ($k = 4$). The CER of k-means clustering on the ORL data is 0.3, which is only improved to about 0.24 by combining feature selection. The limited improvement is due to the non-linear cluster structure. However, directly applying kernel k-means clustering on the full data only leads to a CER 0.23, which implies the potential for further improvement. Moreover, in Example 1.1 of Lymphoma data, although the error rate of clustering of k-means clustering on data without noise features is about 0.05, which can also be further improved if we model the boundaries by non-linear transformations.*

Therefore, it is necessary to develop new methods for clustering high-dimensional data with non-linear cluster structure. From this, we expect to capture more complex cluster structure and to get rid of the negative effects of noise features. Moreover, based on the new proposal, we also hope to realize the extension of kernel clustering to the high-dimensional cases as well.

1.4.2 Missing data

Since the k-means clustering relies on Euclidean distance of each data point to cluster centers, i.e., $\|x_i - \mu_l\|_2$ in Eq. (1.2), the sample (or say data matrix) need to be complete and with no missing values. Otherwise, directly conducting k-means clustering is infeasible.

In the past decades, a vast literature has investigated how to deal with missing data[2]. However, in practical clustering analysis with missing data, the most widely used methods are still naive, for example, deleting the data points including missing values and clustering for the rest data points. Moreover, various imputation methods are also applied for small proportions of missingness, the most simple examples of which are zero-imputation, mean-imputation and regression-imputation. Unfortunately, these traditional methods only work for a small proportion of missingness, and rely on the mechanisms of missingness (that is, assumptions on how the missing values are generated).

In high-dimensional data, the problem of missingness is more ubiquitous, since it is hard to make sure each value of a matrix with millions of entries to be exactly observed. Even if the noise features can be eliminated in the case of including missing values, the rest relevant features may still have missing values and the missingness mechanisms could be complex. In this case, the above traditional methods are no longer effective, or the computational cost is too high to be used in practice. It suggests that except for identifying noise features, we also need to deal with missing data effectively.

For this problem, an existing method called k-POD clustering is effective, even when the missingness proportion is large and the missingness mechanism is unknown (Chi et al. 2016). It reformulates k-means clustering to the matrix decomposition problem with special constraints, and then for the incomplete data matrix, it combines this new formulation with a mapping of the data matrix, such that only observed positions are involved in the objective function. We will introduce the detail in Chapter 3.

However, in high-dimensional data, the effectiveness of the k-POD clustering would be inevitably reduced, due to the existence of plenty of noise

[2] We refer to https://rmisstastic.netlify.app/ for more information.

features. Especially, the estimated cluster centers by k-POD clustering would be biased in the noise features. The main reason is the difference in objective function of k-POD clustering compared to k-means clustering (Terada & Guan 2024). As a consequence, the clustering result would be reduced.

Example 1.3. *In Example 1.1 of Lymphoma data, we consider random missingness of each entry and construct an incomplete data matrix with 30% entries missing. The CER of k-POD clustering on the incomplete data matrix would be 0.3, whereas that of k-means clustering on the complete data matrix after feature selection is only 0.05.*

Therefore, it is necessary to develop new methods for clustering high-dimensional data with missing data. From this, we expect to obtain reasonable estimators of cluster centers when missing data exists. Moreover, based on the new proposal, we also hope to improve the effectiveness of k-POD clustering to the high-dimensional cases as well.

1.5 Aim, objectives and outline

The general aim of this thesis is to make k-means-based clustering methods applicable for high-dimensional data. Specifically, this thesis expects to address the following questions:

- *How to capture the non-linear cluster structure in high-dimensional data with noise features?*

- *How to handle missing values in high-dimensional data with noise features?*

To this end, in this thesis, we will propose two novel clustering methods that cope with the two problems, respectively. The novelties of these two methods are as follows. (1) We combine the kernel k-means clustering with feature selection, so that we can simultaneously find relevant features and recover the non-linear cluster structure they construct. (2) We apply the regularization to the k-POD clustering, so that we can recognize noise features and get reasonable results for missing data clustering.

The main results indicate the effectiveness and better performance of the proposed methods, which will be verified by the experiments on synthetic datasets and applications on real-world datasets. As a consequence, we can extend the application of traditional k-means clustering to more complex data in the big data age.

Throughout this thesis, the number of clusters k is fixed and known, and suppose $k \geq 2$. The number of data dimensions p is large while not change as sample size n increases.

The rest of this thesis is organized as follows.

In Chapter 2, we will address the problem of non-linear cluster structure by proposing sparse kernel k-means clustering. The proposed method assigns each feature a binary indicator to show whether it is selected or not, and then conducts the kernel k-means clustering while penalizing the sum of indicators. The proposed method enables us to capture the ground truth of non-linear cluster structure by the kernel k-means clustering with only selected features. Moreover, it can also be viewed as the extension of kernel k-means clustering to the high-dimensional cases. We would further provide a specific iterative algorithm for optimization and theoretical analysis for consistency. The result of this chapter has been published in *Pattern Recognition* (Guan & Terada 2023).

In Chapter 3, we will address the problem of missingness by proposing regularized k-POD clustering. The proposed method introduces a regularization function of cluster centers to k-POD clustering. By penalizing the cluster centers by features, we are able to get a sparse estimator for cluster centers. It implies that for the high-dimensional missing data, when noise features exist that have no contribution to cluster structure, the proposed method would provide less biased estimators. We would further propose a general framework of optimization for various types of regularization functions. The results of this chapter has been submitted to *Statistics and Computing* and is under the first round of review.

In Chapter 4, we will summarize our contributions, followed by discussing the limitations and possible improvements for future works. For both proposals, numerical experiments and real-world data applications will be conducted to verify the performance of proposed methods. Supplementary details and experiments are provided in Appendix.

Chapter 2

Sparse kernel k-means clustering

2.1 Background

In this chapter, we focus on the high-dimensional data with non-linear cluster structure. In the high-dimensional cases, in most scenarios, the underlying clusters differ in only a small fraction of the features (Witten & Tibshirani 2010), and the redundant features may make traditional clustering methods ineffective.

Conventional clustering methods for high-dimensional data include subspace clustering and dimension reduction-based approaches. However, using these approaches, it is difficult to select relevant features that are highly related to a hidden cluster structure. To find relevant features, one possible way is to compute a clustering relevance measure for each feature and then pick up features with higher scores (Zeng & Cheung 2009, Jin & Wang 2016, Chan & Hall 2010, Zhang et al. 2020). Based on the Gaussian mixture model, a score that measures the dissimilarity between the variance in each cluster and the global variance on each dimension is proposed by Zeng & Cheung (2009) as a relevant feature selection criterion. According to Jin & Wang (2016), Chan & Hall (2010), the relevance measure is characterized by the value of a test statistic used to test for normality. In addition, the score for ranking features used by Zhang et al. (2020) is the amount a feature affects the k-means objective. Another way is to conduct clustering and feature selection simultaneously by adding penalties (Pan & Shen 2007, Wang et al. 2018, Witten & Tibshirani 2010, Chang et al. 2018, Arias-Castro & Pu 2017, Dey et al. 2020). For example, Pan & Shen (2007) considers the Gaussian mixture model and uses the l_1 norm of the mean vector of each component as the penalty term, and Wang et al. (2018) investigates the convex clustering with fused lasso term of the cluster centers. This finally results in a set of sparse cluster centers, and thus the non-zero positions in these cluster centers correspond to relevant features. Witten & Tibshirani (2010) proposes a framework for feature selection in clustering, along with its various extensions (Dey et al. 2020, Chakraborty & Das 2020, Arias-Castro & Pu 2017, Chang et al. 2018, Yang & Benjamin 2023). In this framework, one assigns a weight to each feature and penalizes the sum of the weights when optimizing the objective function of k-means clustering.

However, the k-means clustering has a critical disadvantage in that it cannot capture a complex cluster structure. More specifically, the k-means clustering cannot separate complex clusters with non-linear boundaries, which can be solved by kernel k-means clustering. In kernel k-means clustering, before clustering, the data points are mapped to a higher-dimensional feature space by using a non-linear function, and then kernel k-means clustering partitions the data points by linear separators in the higher-dimensional feature space (Dhillon et al. 2004). As a result, we can capture a non-linear cluster structure in the original input space. However, kernel k-means clustering also faces the curse of dimensionality. Figure 2.1 illustrates this problem. The shape of the points represents the ground truth of the clusters, while the color of the points represents the estimated label. The underlying cluster structure is only generated by the first two features and several irrelevant features are artificially added. As illustrated in left panel, when several features are not relevant to the underlying cluster structure, the kernel k-means clustering with all features fails to capture the reasonable cluster structure. Moreover, the sparse k-means clustering performs poorly to recognize the non-linear cluster structure, as shown in central panel. Therefore, it is important to develop an appropriate feature selection method for the kernel k-means clustering.

There are some difficulties in conducting feature selection for kernel k-means clustering. Since there are no explicit cluster centers in the input space for kernel k-means clustering, we cannot penalize the cluster centers directly to select relevant features, like Pan & Shen (2007), Wang et al. (2018). On the other hand, although sparse clustering framework of Witten & Tibshirani (2010) can be applied, unlike many other extensions of k-means clustering such as Dey et al. (2020), Chakraborty & Das (2020), we cannot derive the explicit expression of the solution of the feature weights. In fact, it requires that the similarity measurement between data points in the entire input space is the sum of that in each dimension, such as the similarity based on the Euclidean norm. Unfortunately, for kernel k-means clustering, the similarity measurement depends on the non-linear mapping before clustering, which

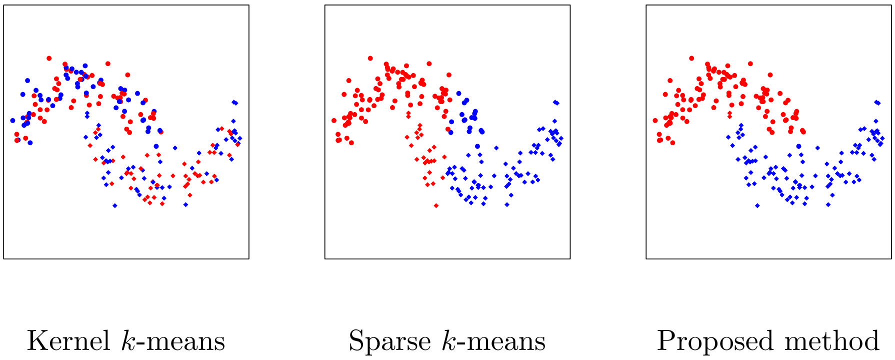


Figure 2.1: A simple example shows how kernel k-means clustering fails in the high-dimensional case. The two illustrated axes are the first two features. The color of each point represents the estimated cluster it is assigned to. The central and right panel are the results of sparse k-means clustering and the proposed method, respectively.

does not satisfy the requirement. Furthermore, although Maldonado et al. (2015) proposes a kernel penalized k-means clustering for feature selection, its objective function is actually based on the soft clustering criterion rather than the kernel k-means clustering itself. Moreover, to our limited knowledge, the theoretical analysis in this area has not yet been fully discussed.

In this chapter, we therefore propose the novel sparse kernel k-means clustering. It assigns a 0-1 indicator to each feature and optimizes an equivalent kernel k-means loss function while penalizing the number of active features. The proposed method can extend the advantages of kernel k-means clustering to the high-dimensional cases. We prove the consistency of both clustering and feature selection of the proposed method under some regularity conditions, and verify the efficacy of the proposed method through detailed experimental studies on several real and synthetic datasets. In addition, we apply the proposed method to normalized cut. Detailed experiments show superior performance.

2.2 Preliminaries for kernel k-means

In this section, we briefly introduce the kernel k-means clustering. It enhances the classical k-means clustering by using an appropriate non-linear mapping from the original data space $\mathcal{X} \subset \mathbb{R}^p$ to a complex space $\mathcal{H}$, so that the non-linear cluster structure in the original space can be extracted

(Dhillon et al. 2004). Moreover, it is usually combined with a non-negative weight w_i for each data point x_i, which is also called weighted kernel k-means clustering (WKKM).

Specifically, for data points $x_1, \ldots, x_n$ in $\mathcal{X}$, denote by $w_i \geq 0$ the corresponding weight for x_i, $i = 1, \ldots, n$. Consider a non-linear mapping $\psi : \mathcal{X} \to \mathcal{H}$, where the space $\mathcal{H}$ is equipped with an inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$. We write $\| \cdot \|_{\mathcal{H}}$ for the norm on $\mathcal{H}$ given by $\|g\|_{\mathcal{H}} = \sqrt{\langle g, g \rangle_{\mathcal{H}}}$ for any $g \in \mathcal{H}$. The objective function of the weighted kernel k-means clustering (WKKM) is given by

$$\widehat{L}_n^{(\mathrm{WKKM})}(\mathcal{C}, \boldsymbol{\mu}) = \frac{1}{n} \sum_{l=1}^{k} \sum_{x_i \in C_l} w_i \|\psi(x_i) - \mu_l\|_{\mathcal{H}}^2, \tag{2.1}$$

where $\mathcal{C} = \{C_1, \ldots, C_k\}$ is a partition of sample, and $\boldsymbol{\mu} = \{\mu_1, \ldots, \mu_k\}$ is the set of centers in $\mathcal{H}$.

The goal of WKKM is to find $(\mathcal{C}, \boldsymbol{\mu})$ that minimizes $\widehat{L}_n^{(\mathrm{WKKM})}$. The optimization is usually solved in a greedy fashion the same as Lloyd's algorithm for the standard k-means clustering, which updates the partition and cluster centers iteratively as follows:

Step 1 Given a partition $\mathcal{C} = \{C_l\}_{l=1}^{k}$, update cluster centers by the "best" cluster representatives of the partition, that is, for $l = 1, \ldots, k$,

$$\mu_l = \frac{\sum_{x_i \in C_l} w_i \psi(x_i)}{\sum_{x_i \in C_l} w_i}.$$

Step 2 Given cluster centers $\boldsymbol{\mu} = \{\mu_l\}_{l=1}^{k}$, update the partition by assigning each data point x_i to its nearest center, that is, for $l = 1, \ldots, k$,

$$C_l = \{x_i, i = 1, \ldots, n \mid \|\psi(x_i) - \mu_l\|_{\mathcal{H}} \leq \|\psi(x_i) - \mu_{l'}\|_{\mathcal{H}}, \forall l' \neq l\}.$$

The choice of $\psi(\cdot)$ is crucial for the clustering effect. It is usually constructed via a kernel function, and this is why it is called *kernel k-means*. Specifically, suppose that $\mathcal{H}$ is a reproducing kernel Hilbert space (RKHS) associated with a reproducing kernel denoted by $h : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. We can define the mapping $\psi : \mathcal{X} \to \mathcal{H}$ by $\psi(x) = h(\cdot, x)$ for any $x \in \mathcal{X}$, which leads to $h(x, \tilde{x}) = \langle \psi(x), \psi(\tilde{x}) \rangle_{\mathcal{H}}$ for any $x, \tilde{x} \in \mathcal{X}$. Then, the objective function of WKKM using kernel h is given by

$$\widehat{L}_n^{(\mathrm{WKKM})}(\mathcal{C}) = \frac{1}{n} \sum_{i=1}^{n} w_i h(x_i, x_i) - \frac{1}{n} \sum_{l=1}^{k} \frac{1}{\sum_{x_i \in C_l} w_i} \sum_{x_i, x_{i'} \in C_l} w_i w_{i'} h(x_i, x_{i'}), \tag{2.2}$$

where we often encode $h_{ii'} = h(x_i, x_{i'})$ for all $i, i' = 1, \ldots, n$ into a kernel matrix $\mathrm{H} = (h_{ii'})_{n \times n}$.

It should be noted that although the mapping $\psi(\cdot)$ is complex, the kernel function h often has simple explicit form (e.g., $h(x, \tilde{x}) = \exp(-\|x - \tilde{x}\|_2^2)$). The formulation of Eq. (2.2) allows us to focus on updating the partition $\mathcal{C}$ without updating $\boldsymbol{\mu}$ explicitly, which means that $\widehat{L}_n^{(\mathrm{WKKM})}$ can be viewed as a function only with respect to $\mathcal{C}$. We provide Algorithm 2.1 for WKKM.

```
Algorithm 2.1 Weighted kernel $k$-means clustering
Input: Kernel matrix H, number of clusters $k$.
  Initialize partition $\mathcal{C}$.
  while $\widehat{L}_n^{(\mathrm{WKKM})}(\mathcal{C})$ does not converge do
    Step 1: For each $i = 1, \ldots, n$, find the index $l^*(i) = \arg\min_{l=1,\ldots,k} d_{il}$, where
      $$d_{il} = h_{ii} - \frac{2}{\sum_{x_{i'} \in C_l} w_{i'}} \sum_{x_{i'} \in C_l} w_{i'} h_{ii'} + \frac{1}{\left(\sum_{x_{i'} \in C_l} w_{i'}\right)^2} \sum_{x_{i'}, x_{i''} \in C_l} w_{i'} w_{i''} h_{i'i''}$$
    Step 2: Update $\mathcal{C} = \{C_l\}_{l=1}^k$ by
      $$C_l = \{x_i, i = 1, \ldots, n \mid l^*(i) = l\}$$
  end while
Output: Partition $\mathcal{C}$.
```

Moreover, the weights $\{w_i\}_{i=1}^n$ are pre-specified, and with some specific weights, the weighted kernel k-means clustering is equivalent to the normalized cut (Ncut) problem (Dhillon et al. 2004, Terada & Yamamoto 2019).

2.3 Proposed method

Suppose that the underlying cluster structure differs only in a subset of features. Our main idea is to assign an indicator to each feature to show whether it is relevant or not. Specifically, we introduce an indicator vector $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_p) \in \{0, 1\}^p$. If $\xi_j = 1$, then the j-th feature is regarded to be relevant, 0 otherwise. Then, the new data $x_1 \circ \boldsymbol{\xi}, \ldots, x_n \circ \boldsymbol{\xi}$ will be used in the weighted kernel k-means clustering, instead of the original data $x_1, \ldots, x_n$, where $\circ$ denotes the element-wised product. Mathematically, we first define $\psi^{\boldsymbol{\xi}} : \mathcal{X} \to \mathcal{H}$ by $\psi^{\boldsymbol{\xi}}(x) = h(\cdot \circ \boldsymbol{\xi}, x \circ \boldsymbol{\xi})$. Then, by substituting $\psi(x_i)$ in Eq.(2.1) by $\psi^{\boldsymbol{\xi}}(x_i)$, it is natural to give the following minimization problem with respect to partition $\mathcal{C}$ and indicator $\boldsymbol{\xi}$:

$$\begin{aligned}&\min_{\mathcal{C},\,\boldsymbol{\xi}} \frac{1}{n}\sum_{l=1}^{k}\sum_{x_i\in C_l} w_i\|\psi^{\boldsymbol{\xi}}(x_i)-\hat{\mu}_l^{\boldsymbol{\xi}}\|_{\mathcal{H}}^2\\ &\text{s.t.}\quad \boldsymbol{\xi}\in\{0,1\}^p \text{ and } \|\boldsymbol{\xi}\|_0\le d,\end{aligned}\tag{2.3}$$

where for all $l=1,\dots,k$,

$$\hat{\mu}_l^{\boldsymbol{\xi}} = \frac{\sum_{x_i\in C_l} w_i\psi^{\boldsymbol{\xi}}(x_i)}{\sum_{x_i\in C_l} w_i}.$$

Denote this objective function by WCSS$(\mathcal{C},\boldsymbol{\xi})$. Then, minimizing WCSS$(\mathcal{C},\boldsymbol{\xi})$ with a fixed $\mathcal{C}$ would lead to a trivial solution $\hat{\boldsymbol{\xi}}=0$.

Example 2.1. *Consider $\psi(\cdot)$ constructed by the kernel function $h(x,\tilde{x}) = \exp(-\|x-\tilde{x}\|_2^2)$ and let $w_i=1$ for all $i=1,\dots,n$. Then we have*

$$WCSS(\mathcal{C},\boldsymbol{\xi}) = 1-\frac{1}{n}\sum_{l=1}^{k}\frac{1}{|C_l|}\sum_{x_i,x_{i'}\in C_l}\exp\left(-\|(x_i-x_{i'})\circ\boldsymbol{\xi}\|_2^2\right),$$

which implies that $\hat{\boldsymbol{\xi}}=0$ is a minimizer of WCSS$(\mathcal{C},\boldsymbol{\xi})$ when $\mathcal{C}$ is fixed.

To avoid the trivial solution, we consider a contrast expression and propose the sparse (weighted) kernel k-means clustering (SKKM) as follows:

$$\begin{aligned}&\max_{\mathcal{C},\,\boldsymbol{\xi}}\left\{\frac{1}{n}\sum_{i=1}^{n} w_i\|\psi^{\boldsymbol{\xi}}(x_i)-\hat{\mu}_0^{\boldsymbol{\xi}}\|_{\mathcal{H}}^2-\frac{1}{n}\sum_{l=1}^{k}\sum_{x_i\in C_l} w_i\|\psi^{\boldsymbol{\xi}}(x_i)-\hat{\mu}_l^{\boldsymbol{\xi}}\|_{\mathcal{H}}^2\right\}\\ &\text{s.t. } \boldsymbol{\xi}\in\{0,1\}^p \text{ and } \|\boldsymbol{\xi}\|_0\le d,\end{aligned}\tag{2.4}$$

where for all $l=1,\dots,k$,

$$\hat{\mu}_0^{\boldsymbol{\xi}} = \frac{\sum_{i=1}^{n} w_i\psi^{\boldsymbol{\xi}}(x_i)}{\sum_{i=1}^{n} w_i} \text{ and } \hat{\mu}_l^{\boldsymbol{\xi}} = \frac{\sum_{x_i\in C_l} w_i\psi^{\boldsymbol{\xi}}(x_i)}{\sum_{x_i\in C_l} w_i}.$$

Denote the objective function by BCSS$(\mathcal{C},\boldsymbol{\xi})$. There are two advantages of Eq.(2.4):

(1) Since the first term does not rely on the partition $\mathcal{C}$ (thus denoted by TSS$(\boldsymbol{\xi})$) and the second term is actually WCSS$(\mathcal{C},\boldsymbol{\xi})$, then for a given indicator $\boldsymbol{\xi}$, maximizing BCSS$(\mathcal{C})$ is equivalent to minimizing WCSS$(\mathcal{C})$ and thus they provide the same clustering result.

(2) For a given partition $\mathcal{C}$, since maximizing BCSS($\boldsymbol{\xi}$) implies not only minimizing WCSS($\boldsymbol{\xi}$) but also maximizing TSS($\boldsymbol{\xi}$), it helps to avoid the trivial solution and thus provides a reasonable result for feature selection.

Moreover, the relationship of WCSS, BCSS and TSS can be explained as that the distortion of the whole sample (TSS) can be decomposed by the distortion of sample within k clusters (WCSS) and the distortion between k clusters (BCSS). The similar trick is also used by Witten & Tibshirani (2010), Dey et al. (2020).

It should be noted that the proposed method is different from Maldonado et al. (2015). The indicator $\boldsymbol{\xi}$ used in our method is assumed to be binary, while the similar parameter used in their method is continuous and meanwhile serves as the bandwidth of the kernel function. The standard objective function of (weighted) kernel k-means clustering is not considered in their method, whereas directly involved in our method. Moreover, it should be noted that the weights used in our method are assigned to data points, which is aimed to improve the effect of clustering. However, the weights used by Witten & Tibshirani (2010), Chang et al. (2018), Dey et al. (2020), Chakraborty & Das (2020) are assigned to features, which play the similar role as $\boldsymbol{\xi}$ used in our method.

In addition to Eq. (2.4), we also propose an equivalent expression of SKKM based on the kernel function h as follows, which facilitates the implementation:

$$\max_{\mathcal{C},\,\boldsymbol{\xi}} \left\{ \frac{1}{n} \sum_{l=1}^{k} \frac{1}{\sum_{x_i \in C_l} w_i} \sum_{x_i, x_{i'} \in C_l} w_i w_{i'} h^{\boldsymbol{\xi}}_{i,i'} - \frac{1}{n \sum_{i=1}^{n} w_i} \sum_{i,i'=1}^{n} w_i w_{i'} h^{\boldsymbol{\xi}}_{i,i'} \right\} \tag{2.5}$$
$$\text{s.t. } \boldsymbol{\xi} \in \{0,1\}^p \;\text{ and }\; \|\boldsymbol{\xi}\|_0 \le d,$$

where $h^{\boldsymbol{\xi}}_{i,i'} = h(x_i \circ \boldsymbol{\xi}, x_{i'} \circ \boldsymbol{\xi})$ for any $i, i' = 1, \ldots, n$, which is also encoded into a matrix $\mathrm{H}^{\boldsymbol{\xi}} = (h^{\boldsymbol{\xi}}_{i,i'})_{n \times n}$.

Remark 2.1. *The motivation for using the l_0 constraint of $\boldsymbol{\xi}$ is due to the consideration of practical optimization. To solve the l_0 regularized optimization problem, a common way is to relax l_0 to l_1 so that the NP-hard problem can be avoided. That is, the constraint $\boldsymbol{\xi} \in \{0,1\}^p$ and $\|\boldsymbol{\xi}\|_0 \le d$ is relaxed to $\boldsymbol{\xi} \in [0,1]^p$ and $\|\boldsymbol{\xi}\|_1 \le d$. Then, the original l_0 solution could be given by taking the sign of the relaxed l_1 solution. However, in our case, due to the non-convexity of the objective function, we can hardly benefit much from the l_1 relaxation. In fact, the relaxed version is a constrained non-convex optimization problem, solving which is quite challenging for the existing well-built* *solvers. Moreover, even for the l_1 penalty function, there are several well-known issues (e.g., see Fan & Li (2001)). Therefore, our algorithm focuses on directly finding a local solution to the original l_0 optimization problem as introduced in the next section.*

2.4 Optimization

2.4.1 Algorithms

For implementation, similar to WKKM, we consider the maximization problem of Eq.(2.5) with respect to partition $\mathcal{C}$ and indicator $\boldsymbol{\xi}$, where a pre-specified kernel function h is used, so that we can avoid updating cluster centers explicitly. Then, since the objective function is generally non-convex, we consider the alternative maximization between $\boldsymbol{\xi}$ and $\mathcal{C}$ until convergence, and propose Algorithm 2.2 to obtain a local solution for SKKM.

Specifically, for the $(t+1)$-th iteration, in Step 1, we keep $\mathcal{C}^{(t)}$ fixed and find a maximizer $\boldsymbol{\xi}^{(t+1)}$ of the objective function, which is denoted by $f_n(\boldsymbol{\xi} \mid \mathcal{C}^{(t)})$ here for short. That is, our goal in this step is to maximize $f_n(\boldsymbol{\xi} \mid \mathcal{C}^{(t)})$ with $\boldsymbol{\xi} \in \{0,1\}^p$ and $\|\boldsymbol{\xi}\|_0 \leq d$. To this end, it suffices to find the optimal support set S of $\boldsymbol{\xi}$ with the cardinality $|S| \leq d$. Therefore, we consider a stepwise method as follows to find S. We start from the empty set $S = \emptyset$, and repeat the following until the $|S| = d$:

(a) Find $j^* = \underset{j \in [p] \setminus S}{\arg\max} f_n(\boldsymbol{e}_{S \cup \{j\}} \mid \mathcal{C}^{(t)})$, where $\boldsymbol{e}_\Omega \in \mathbb{R}^p$ is the 0-1 valued vector supported on the index set Ω, and $[p] = \{1, \ldots, p\}$;

(b) Update S by adding j^* into it.

Then we can get $\boldsymbol{\xi}^{(t+1)}$ by taking its j-th component $\xi_j^{(t+1)} = 1$ if $j \in S$, 0 otherwise.

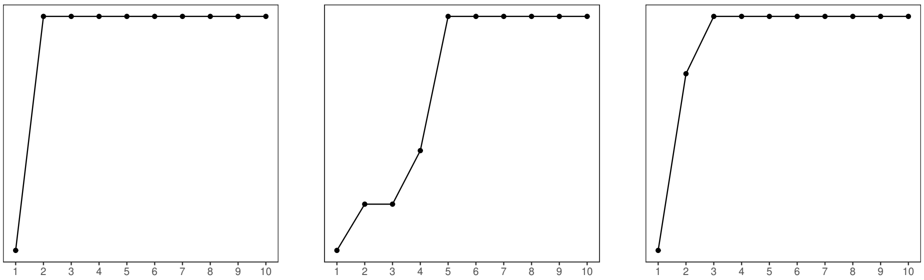
In Step 2, we keep $\boldsymbol{\xi}^{(t+1)}$ fixed and find the optimal partition $\mathcal{C}^{(t+1)}$. Here the objective function is equivalent to that of the classical weighted kernel k-means clustering on the new sample $\{x_1 \circ \boldsymbol{\xi}^{(t+1)}, \ldots, x_n \circ \boldsymbol{\xi}^{(t+1)}\}$. Therefore, we calculate a new kernel matrix $\mathrm{H}^{\boldsymbol{\xi}^{(t+1)}} = (h_{ii'}^{\boldsymbol{\xi}^{(t+1)}})_{n \times n}$ with elements being $h_{ii'}^{\boldsymbol{\xi}^{(t+1)}} = h(x_i \circ \boldsymbol{\xi}^{(t+1)}, x_{i'} \circ \boldsymbol{\xi}^{(t+1)})$, for all $i, i' = 1, \ldots, n$. Then, update $\mathcal{C}^{(t+1)}$ by conducting Algorithm 2.1 with the input $\mathrm{H}^{\boldsymbol{\xi}^{(t+1)}}$ and k.

It should be noted that when the weighted version is used, the weights $\{w_1, \ldots, w_n\}$ should be pre-specified as an input, which is omitted in Algorithm 2.2 for the simplification of notations.

Algorithm 2.2 Sparse (weighted) kernel k-means clustering

```
Input: Sample $\{x_1, \ldots, x_n\}$, number of clusters $k$, kernel function $h$.
Parameters: Number of active features $d$.
  Initialize partition $\mathcal{C}^{(0)}$.
  while not converge do
      Step 1: Keeping $\mathcal{C}^{(t)}$ fixed, solve Eq. (2.5) w.r.t. $\boldsymbol{\xi}$ as follows:
          Let $S = \emptyset$.
          while $|S| < d$ do
              (a) Find   $j^* = \arg\max_{j \in [p] \setminus S} f_n(\boldsymbol{e}_{S \cup \{j\}} \mid \mathcal{C}^{(t)})$;
              (b) Update $S \leftarrow S \cup \{j^*\}$.
          end while
          Update $\boldsymbol{\xi}^{(t+1)} = (\xi_1^{(t+1)}, \ldots, \xi_p^{(t+1)})$ by
              $$\xi_j^{(t+1)} = \begin{cases} 1 & \text{if } j \in S \\ 0 & \text{otherwise} \end{cases}$$
      Step 2: Keeping $\boldsymbol{\xi}^{(t+1)}$ fixed, solve Eq. (2.5) w.r.t. $\mathcal{C}$ as follows:
          Apply Algorithm 2.1 with input $\mathrm{H}^{\boldsymbol{\xi}^{(t+1)}} = (h_{ii'}^{\boldsymbol{\xi}^{(t+1)}})_{n \times n}$, where
              $$h_{ii'}^{\boldsymbol{\xi}^{(t+1)}} = h(x_i \circ \boldsymbol{\xi}^{(t+1)}, x_{i'} \circ \boldsymbol{\xi}^{(t+1)});$$
          Update $\mathcal{C}^{(t+1)}$ by the output.
  end while
Output: Partition $\mathcal{C}^{(t+1)}$, indicator $\boldsymbol{\xi}^{(t+1)}$.
```

Moreover, as for the convergence issue, we iterate Step 1 and Step 2 until the update of $\boldsymbol{\xi}$ does not change. According to our numerical experiments, for most datasets, the stop criterion would be met within several times of iterations. In Figure 2.2, we empirically demonstrate the gradual convergence of the proposed algorithm to a local stationary point on several selected datasets. For other datasets, the convergence trends are similar. We further note that the number of iterations to convergence is generally related to the complexity of the clustering problem. For a more complicated clustering problem, more iterations are needed. It is also worth noting that the immediate convergence may be the common characteristic of the alternative algorithms for the simultaneous clustering and feature selection issue, as mentioned in other related works (Dey et al. 2020, Chakraborty & Das 2020). For the sake of space economy, we have a detailed discussion in Section A.1 of Appendix A.



Brain Colon Leukemia

Figure 2.2: Examples of empirical convergence of the proposed algorithm. The x-axis is the index of iteration. The y-axis is the value of objective function of each iteration. Each panel shows the result of each dataset.

In addition, we analyze the complexity of the proposed algorithm. In Step 1, the stepwise method has the complexity $O(n^2pd)$. In Step 2, the complexity of constructing the kernel matrix is $O(n^2p)$ while the complexity of estimating partition by WKKM is $O(n^2\tau)$, where τ is the total number of iterations within WKKM. Thus, Step 2 has a complexity of $O(n^2(p+\tau))$. Therefore, the asymptotic complexity of each iteration of the proposed algorithm is nearly $O(n^2pd)$.

2.4.2 Selection of tuning parameter

To apply the proposed method, we need to pre-specify a kernel function h, the number of active features d and the weights $\{w_i\}_{i=1}^n$ if needed.

For the kernel function, we generally consider the form $h(x, \tilde{x}) = \phi(\|x - \tilde{x}\|_2/\nu)$, where $\phi : \mathbb{R}_+ \to \mathbb{R}_+$ is a function, $\nu \geq 0$ is the bandwidth. In our experiments, we mainly consider $\phi(t) = \exp(-t^2)$, and use the corresponding Gaussian kernel

$$h(x, \tilde{x}) = \exp(-\|x - \tilde{x}\|_2^2/\nu^2)$$

The bandwidth is chosen by using an empirical formula called the median heuristic (Garreau et al. 2017, Paul et al. 2022),

$$\nu^2 = \frac{1}{2}\text{Median}\{\|x_i - x_{i'}\|_2^2 \mid 1 \leq i < i' \leq n\}. \tag{2.6}$$

According to our experiments in Section 2.6.5, we found that Eq. (2.6) performs better compared with other smaller scales of it.

For the number of active features, we use gap statistics to be the criterion, which measures the change in within-clusters dispersion (WCSS) of original data with that expected under null data that does not contain subgroups (Tibshirani et al. 2002). In our case, we focus on the change in between-clusters dispersion (BCSS). The calculation of the gap statistics for a candidate d is as follows:

Step 1 For the original data matrix X, calculate $\widehat{\text{BCSS}}$, that is, the objective functions Eq. (2.5) obtained by applying the proposed method on X.

Step 2 Generate M permuted data matrices $\text{X}(1), \ldots, \text{X}(M)$ by independently permuting observations in each feature.

Step 3 For each $\text{X}(m)$ $(m = 1, \ldots, M)$, calculate $\widehat{\text{BCSS}}_m$, i.e., the objective functions Eq. (2.5) obtained by applying the proposed method on $\text{X}(m)$.

Step 4 Calculate the gap statistics for the candidate d by

$$\text{Gap}(d) = \log(\widehat{\text{BCSS}}) - \frac{1}{M}\sum_{m=1}^{M} \log(\widehat{\text{BCSS}}_m).$$

The candidate d with the largest gap statistics is selected. In our experiments, the "within standard deviation" trick is used to avoid trivial choices and obtain a reasonable subset of features. It should be noted that although the gap statistics was initially proposed to estimate the number of clusters, it has been widely used by Witten & Tibshirani (2010), Dey et al. (2020), Chang et al. (2018) for other tuning parameters of clustering problem.

For the weights (if needed), in order to apply the proposed method to Ncut, based on the relationship between WKKM and Ncut (Dhillon et al. 2004), we construct the weights for the proposed method such that if all features are selected, SKKM coincides with Ncut. The specific procedure is provided in Section A.3.1 of Appendix A.

2.5 Theoretical results

2.5.1 Notations and assumptions

Let $\mathcal{X}$ be a compact metric data space, $\mathcal{X} \subset \mathbb{R}^p$, and let $\mathbb{P}$ be a probability measure, the support of which is $\mathcal{X}$. Denote by $\{X_1, \ldots, X_n\}$ a random sample in $\mathcal{X}$, where X_i's are independent random vectors and identically distributed following $\mathbb{P}$. For any $x, \tilde{x} \in \mathcal{X}$ with j-th component given by $x_j, \tilde{x}_j$, the inner product is $\langle x, \tilde{x} \rangle_2 = \sum_{j=1}^p x_j \tilde{x}_j$ and the norm is given by $\|x\|_2 = \sqrt{\sum_{j=1}^p x_j^2}$.

The function $h : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a reproducing kernel. Let $\mathcal{H}$ be the corresponding reproducing kernel Hilbert space (RKHS). On the space $\mathcal{H}$, the inner product is denoted by $\langle \cdot, \cdot \rangle_{\mathcal{H}}$, which defines a norm $\|\cdot\|_{\mathcal{H}}$, that is, for any $g \in \mathcal{H}$, $\|g\|_{\mathcal{H}} = \sqrt{\langle g, g \rangle_{\mathcal{H}}}$. In our method, we define the mapping $\psi : \mathcal{X} \to \mathcal{H}$ by $\psi(x) = h(\cdot, x)$ for any $x \in \mathcal{X}$, and we have $h(x, \tilde{x}) = \langle \psi(x), \psi(\tilde{x}) \rangle_{\mathcal{H}}$ for any $x, \tilde{x} \in \mathcal{X}$. Assume that h is bounded by $h(x, \tilde{x}) \leq c_U$ for any $x, \tilde{x} \in \mathcal{X}$ and $\mathbb{E}_{X \sim \mathbb{P}}[h(X, X)] < \infty$.

Moreover, we write $\boldsymbol{\mu} = \{\mu_1, \ldots, \mu_k\}$, where $\mu_l \in \mathcal{H}$ is the l-th cluster center $(l = 1, \ldots, k)$ in $\mathcal{H}$. We use $\boldsymbol{\mu} \in \mathcal{H}_k$ to express $\mu_l \in \mathcal{H}$ for all $l = 1, \ldots, k$. We assume the indicator $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_p) \in \{0, 1\}^p$. Then, for any such $\boldsymbol{\xi}$, we define $h^{\boldsymbol{\xi}} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ by $h^{\boldsymbol{\xi}}(x, \tilde{x}) = h(x \circ \boldsymbol{\xi}, \tilde{x} \circ \boldsymbol{\xi})$. The corresponding RKHS is denoted by $\mathcal{H}^{\boldsymbol{\xi}}$. We further define $\psi^{\boldsymbol{\xi}} : \mathcal{X} \to \mathcal{H}^{\boldsymbol{\xi}}$ by $\psi^{\boldsymbol{\xi}}(x) = h(\cdot \circ \boldsymbol{\xi}, x \circ \boldsymbol{\xi})$, and we write the corresponding sample average $\hat{\mu}_0^{\boldsymbol{\xi}}$ and population mean $\mu_0^{\boldsymbol{\xi}}$ to be

$$\hat{\mu}_0^{\boldsymbol{\xi}} = \frac{1}{n} \sum_{i=1}^n \psi^{\boldsymbol{\xi}}(X_i) \quad \text{and} \quad \mu_0^{\boldsymbol{\xi}} = \mathbb{E}_{X \sim \mathbb{P}}[\psi^{\boldsymbol{\xi}}(X)].$$

Throughout the theoretical analysis, we suppose the number of clusters $k \geq 2$ and the number of relevant features d_0 to be $1 \leq d_0 < p$, both of which are fixed. Moreover, for the sake of notation simplicity, we focus on the unweighted version of the proposed method, that is, we fix $w_i = 1$ for all $i = 1, \dots, n$. We can easily extend the proof for the general weighted case. To facilitate the derivation, we focus on kernel function, associated with which, the RKHS satisfies the following assumption:

Assumption 2.1. *For any $\boldsymbol{\eta}, \tilde{\boldsymbol{\eta}} \in \{0,1\}^p$, if the support of $\boldsymbol{\eta}$ is included in the support of $\tilde{\boldsymbol{\eta}}$, that is, $\{j = 1, \dots, p \mid \eta_j = 1\} \subset \{j = 1, \dots, p \mid \tilde{\eta}_j = 1\}$, then we have $\mathcal{H}^{\boldsymbol{\eta}} \subset \mathcal{H}^{\tilde{\boldsymbol{\eta}}}$.*

It implies that for any $\boldsymbol{\eta} \in \{0,1\}^p$, we have $\mathcal{H}^{\boldsymbol{\eta}} \subset \mathcal{H}$. In addition, since the exponential kernel with the form of $h(x, \tilde{x}) = \exp(\langle x, \tilde{x} \rangle_2)$ satisfies this assumption, then in our analysis, we will take this special case of kernel functions as an example. We provide some preliminaries about exponential kernel in Section 2.7.1.

Furthermore, let $\mathbf{1}_p = (1, \dots, 1)$ be the all-one vector in $\mathbb{R}^p$. Define

$$D(\boldsymbol{\mu}, \tilde{\boldsymbol{\mu}}) = \max\left\{ \max_{l'=1,\dots,k} \min_{l=1,\dots,k} \|\mu_l - \tilde{\mu}_{l'}\|_{\mathcal{H}}, \max_{l=1,\dots,k} \min_{l'=1,\dots,k} \|\mu_l - \tilde{\mu}_{l'}\|_{\mathcal{H}} \right\},$$

and write $\Theta^* = \{\boldsymbol{\xi} \in \{0,1\}^p \mid \xi_j = 1, \forall j = 1, \dots, d_0\}$ and $\boldsymbol{\xi}^{**} = (\mathbf{1}_{d_0}, \mathbf{0}_{p-d_0})$.

2.5.2 Reformulation

Firstly, we reformulate the proposed method (SKKM) given in Eq. (2.4) to be a maximization problem with respect to $\boldsymbol{\mu}$ and $\boldsymbol{\xi}$. Consider the correspondence between cluster centers $\boldsymbol{\mu} = \{\mu_1, \dots, \mu_k\}$ and partition $\mathcal{C} = \{C_1, \dots, C_k\}$:

(1) For a fixed partition $\mathcal{C}$, cluster centers $\boldsymbol{\mu}$ are given by $\mu_l = \sum_{X_i \in C_l} \psi^{\boldsymbol{\xi}}(X_i) / |C_l|$;

(2) For fixed cluster centers $\boldsymbol{\mu}$, the partition $\mathcal{C}$ is given by $C_l = \tilde{C}_l \setminus \bigcup_{l'<l} \tilde{C}_{l'}$, where $\tilde{C}_l = \{X_i \mid \|\psi^{\boldsymbol{\xi}}(X_i) - \mu_l\|_{\mathcal{H}} \le \|\psi^{\boldsymbol{\xi}}(X_i) - \mu_{l'}\|_{\mathcal{H}},\ \forall l' \neq l\}$.

Therefore, it allows us to rewrite SKKM as a maximization problem with respect to $\boldsymbol{\mu}$ and $\boldsymbol{\xi}$. Specifically, define $\widetilde{L}_n^{(\mathrm{SKKM})}(\cdot,\cdot) : \mathcal{H}_k \times \{0,1\}^p \to \mathbb{R}$ to be

$$\widetilde{L}_n^{(\mathrm{SKKM})}(\boldsymbol{\mu}, \boldsymbol{\xi}) = \frac{1}{n} \sum_{i=1}^{n} \left\{ \|\psi^{\boldsymbol{\xi}}(X_i) - \hat{\mu}_0^{\boldsymbol{\xi}}\|_{\mathcal{H}}^2 - \min_{l=1,\dots,k} \|\psi^{\boldsymbol{\xi}}(X_i) - \mu_l\|_{\mathcal{H}}^2 \right\}.$$

The SKKM with respect to $\boldsymbol{\mu} = \{\mu_1, \dots, \mu_k\}$ and $\boldsymbol{\xi} \in \{0,1\}^p$ is given by

$$\max_{\substack{\boldsymbol{\xi} \in \{0,1\}^p \\ \mu_l \in \mathcal{H}^{\boldsymbol{\xi}},\ \forall l=1,\dots,k}} \widetilde{L}_n^{(\mathrm{SKKM})}(\boldsymbol{\mu}, \boldsymbol{\xi}) \quad \text{s.t.} \quad \|\boldsymbol{\xi}\|_0 \le d, \tag{2.7}$$

where $d = 1, \ldots, p$ is a user-specified number of active features[1]. We denote the maximizer of Eq. (2.7) by $(\tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\xi}})$.

Secondly, since Eq. (2.7) is a constrained optimization problem, we further transform it to be an unconstrained problem with a penalty term. Let $\lambda_n > 0$ be a constant relying on n. We define $\widehat{L}_n(\cdot, \cdot) : \mathcal{H}_k \times \{0,1\}^p \to \mathbb{R}$,

$$\widehat{L}_n(\boldsymbol{\mu}, \boldsymbol{\xi}) = \frac{1}{n} \sum_{i=1}^{n} \left\{ \|\psi^{\boldsymbol{\xi}}(X_i) - \hat{\mu}_0^{\boldsymbol{\xi}}\|_{\mathcal{H}}^2 - \min_{l=1,\ldots,k} \|\psi^{\boldsymbol{\xi}}(X_i) - \mu_l\|_{\mathcal{H}}^2 \right\} - \lambda_n \|\boldsymbol{\xi}\|_0. \tag{2.8}$$

The penalized version of the SKKM can be given by

$$\max_{\substack{\boldsymbol{\xi} \in \{0,1\}^p \\ \mu_l \in \mathcal{H}^{\boldsymbol{\xi}}, \forall l = 1,\ldots,k}} \widehat{L}_n(\boldsymbol{\mu}, \boldsymbol{\xi}), \tag{2.9}$$

and then the estimator of cluster centers and indicator is given by

$$(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\xi}}) = \operatorname*{arg\,max}_{\substack{\boldsymbol{\xi} \in \{0,1\}^p \\ \mu_l \in \mathcal{H}^{\boldsymbol{\xi}}, \forall l = 1,\ldots,k}} \widehat{L}_n(\boldsymbol{\mu}, \boldsymbol{\xi}). \tag{2.10}$$

Proposition 2.1. *The optimization problems of Eq. (2.7) and Eq. (2.9) are equivalent, if there exists* $\lambda_n > 0$ *such that*

$$\begin{aligned} &\lambda_n \geq \max_{\|\boldsymbol{\xi}\|_0 > d} \frac{\widetilde{L}_n^{(\mathrm{SKKM})}(\tilde{\boldsymbol{\mu}}^{\boldsymbol{\xi}}, \boldsymbol{\xi}) - \widetilde{L}_n^{(\mathrm{SKKM})}(\tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\xi}})}{\|\boldsymbol{\xi}\|_0 - d} \\ \text{and} \quad &\lambda_n \leq \min_{\|\boldsymbol{\xi}\|_0 < d} \frac{\widetilde{L}_n^{(\mathrm{SKKM})}(\tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\xi}}) - \widetilde{L}_n^{(\mathrm{SKKM})}(\tilde{\boldsymbol{\mu}}^{\boldsymbol{\xi}}, \boldsymbol{\xi})}{d - \|\boldsymbol{\xi}\|_0}, \end{aligned} \tag{2.11}$$

where $\tilde{\boldsymbol{\mu}}^{\boldsymbol{\xi}} = \operatorname*{arg\,max}_{\boldsymbol{\mu} \in \mathcal{H}_k^{\boldsymbol{\xi}}} \widetilde{L}_n^{(\mathrm{SKKM})}(\boldsymbol{\mu}, \boldsymbol{\xi})$ *for any fixed* $\boldsymbol{\xi} \in \{0,1\}^p$.

Proof. The proof is provided in Section 2.7.2. □

On one hand, for the penalized version with λ_n, we can always ensure that there exists some $d = 1, \ldots, p$ such that the constrained version with d is equivalent to the penalized version. On the other hand, for the constrained version with d, it is equivalent to some penalized version if the associated λ_n satisfying Eq. (2.11) exists.

[1] In contrast, d_0 is a fixed value representing the true number of relevant features in the theoretical analysis.

Remark 2.2. *The existence of λ_n can be ensured under certain conditions for specific values of d. When d is exactly the true value d_0, or when d is chosen to be an empirical estimator $\tilde{d}_n$ that depends on the sample and approaches to the true value d_0 as $n \to \infty$, then any λ_n with $\lim_{n\to\infty} \lambda_n = 0$ and $\lim_{n\to\infty} \lambda_n \sqrt{n} = \infty$ satisfies the condition (2.11). More precisely, for such a sequence of λ_n, the probability that the condition (2.11) holds converges to one. As an example of empirical $\tilde{d}_n$, we can consider the following estimator:*

$$\tilde{d}_n = \min\{t \in \{1, \dots, p\} \mid \widetilde{Q}_n(t) > \max_{s=1,\dots,p} \widetilde{Q}_n(s) - \gamma_n\}, \tag{2.12}$$

where $\widetilde{Q}_n(t) = \max\left\{\widetilde{L}_n^{(\mathrm{SKKM})}(\boldsymbol{\mu}, \boldsymbol{\xi}) \mid \boldsymbol{\xi} \in \{0,1\}^p, \|\boldsymbol{\xi}\|_0 \leq t, \boldsymbol{\mu} \in \mathcal{H}_k^{\boldsymbol{\xi}}\right\}$, and $\{\gamma_n\}_{n\in\mathbb{N}}$ is a sequence with $\lim_{n\to\infty} \gamma_n = 0$ and $\lim_{n\to\infty} \gamma_n\sqrt{n} = \infty$. We can ensure that this estimator $\tilde{d}_n$ converges to the true value d_0 in probability. We leave more discussions and proofs in Section 2.7.7 for the sake of space.

Consequently, Proposition 2.1 allows us to focus on the penalized version of SKKM (Eq. (2.9)), and analyze properties of its maximizer $(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\xi}})$.

2.5.3 Main results

Our aim in this section is to analyze the convergence of the estimated cluster centers and indicator $(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\xi}})$ given in Eq.(2.10) based on the penalized version of SKKM.

First, as a counterpart of $\widehat{L}_n$ in the population level, we define $L(\cdot,\cdot) : \mathcal{H}_k \times \{0,1\}^p \to \mathbb{R}$,

$$L(\boldsymbol{\mu}, \boldsymbol{\xi}) = \mathbb{E}_{X\sim\mathbb{P}}\left[\|\psi^{\boldsymbol{\xi}}(X) - \mu_0^{\boldsymbol{\xi}}\|_{\mathcal{H}}^2 - \min_{l=1,\dots,k} \|\psi^{\boldsymbol{\xi}}(X) - \mu_l\|_{\mathcal{H}}^2\right]. \tag{2.13}$$

As a counterpart of $(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\xi}})$, the optimal solution in the population level is given by

$$(\boldsymbol{\mu}^*, \boldsymbol{\xi}^*) \in \mathop{\arg\max}_{\substack{\boldsymbol{\xi}\in\{0,1\}^p \\ \mu_l \in \mathcal{H}^{\boldsymbol{\xi}}, \forall l=1,\dots,k}} L(\boldsymbol{\mu}, \boldsymbol{\xi}). \tag{2.14}$$

In addition, we define the population mean in $\mathcal{H}$ by $\mu_0 = \mathbb{E}_{X\sim\mathbb{P}}[\psi(X)]$. Then, the optimal cluster centers given by kernel k-means using all features can be expressed as

$$\boldsymbol{\mu}^{**} = \mathop{\arg\max}_{\boldsymbol{\mu}\in\mathcal{H}_k} L(\boldsymbol{\mu}, \mathbf{1}_p). \tag{2.15}$$

Condition 2.1 (Irrelevance). *The $\boldsymbol{\mu}^{**} = \{\mu_1^{**}, \ldots, \mu_k^{**}\}$ is unique and each $\mu_l^{**} : \mathcal{X} \to \mathbb{R}$ $(l = 1, \ldots, k)$ satisfies that $\mu_l^{**}(x)$ only relies on $x_1, \ldots, x_{d_0}$. That is, for any $x, \tilde{x} \in \mathcal{X}$, if $x_j = \tilde{x}_j$ for all $j = 1, \ldots, d_0$, then $\mu_l^{**}(x) = \mu_l^{**}(\tilde{x})$.*

Condition 2.2 (Independence). *The random vector X_1 following the distribution $\mathbb{P}$ satisfies the independence $(X_{11}, \ldots, X_{1d_0}) \perp\!\!\!\perp (X_{1(d_0+1)}, \ldots, X_{1p})$. Moreover, the random variables $X_{1(d_0+1)}, \ldots, X_{1p}$ following non-degenerated distributions $\mathbb{P}_{d_0+1}, \ldots, \mathbb{P}_p$ are independent.*

Condition 2.3 (Normalization). *Suppose that for any $\boldsymbol{\xi} \in \Theta^*$, it holds that*

$$\mathbb{E}_{X\sim\mathbb{P}}\mathbb{E}_{\tilde{X}\sim\mathbb{P}}\big[h(X \circ \boldsymbol{\xi}^{**}, \tilde{X} \circ \boldsymbol{\xi}^{**})\big] \leq \mathbb{E}_{X\sim\mathbb{P}}\mathbb{E}_{\tilde{X}\sim\mathbb{P}}\big[h(X \circ \boldsymbol{\xi}, \tilde{X} \circ \boldsymbol{\xi})\big],$$

where $\tilde{X}$ is a random vector independent to X.

Condition 2.4 (Optimality). *Suppose that for any $\boldsymbol{\xi} \notin \Theta^*$, it holds that*

$$L(\boldsymbol{\mu}^{**}, \boldsymbol{\xi}^{**}) > \sup\{L(\boldsymbol{\mu}, \boldsymbol{\xi}) \mid \boldsymbol{\xi} \in \{0,1\}^p, \boldsymbol{\xi} \notin \Theta^*; \mu_l \in \mathcal{H}^{\boldsymbol{\xi}}, \forall l = 1, \ldots, k\}.$$

The above conditions provide a sparse cluster structure with d_0 relevant features and $p - d_0$ noise features. The Conditions 2.1 and 2.2 are to specify d_0 relevant features and $p - d_0$ noise features. The Conditions 2.3 and Condition 2.4 are technical conditions for the following Proposition 2.2, which specifies maximizers of $L(\cdot, \cdot)$. Based on these conditions, we can naturally regard $\boldsymbol{\mu}^{**}$ and $\boldsymbol{\xi}^{**}$ as the "true" cluster centers and "true" indicator vector.

Proposition 2.2. *Under Conditions 2.1-2.4, any maximizer $(\boldsymbol{\mu}^*, \boldsymbol{\xi}^*)$ of $L(\boldsymbol{\mu}, \boldsymbol{\xi})$ satisfies*

$$\boldsymbol{\mu}^* = \boldsymbol{\mu}^{**} \text{ and } \boldsymbol{\xi}^* \in \Theta^*.$$

Moreover, $(\boldsymbol{\mu}^{**}, \boldsymbol{\xi}^{**})$ is one of maximizers of $L(\boldsymbol{\mu}, \boldsymbol{\xi})$. In addition, when h is the exponential kernel, $(\boldsymbol{\mu}^{**}, \boldsymbol{\xi}^{**})$ is the unique maximizer of $L(\boldsymbol{\mu}, \boldsymbol{\xi})$.

Proof. The proof is provided in Section 2.7.3. □

Theorem 2.1. *Under Conditions 2.1-2.4, and assume $\lim_{n\to\infty} \lambda_n = 0$, then*

(i) For any $\tilde{\epsilon} > 0$, we have $\lim_{n\to\infty} \Pr\left(L(\boldsymbol{\mu}^*, \boldsymbol{\xi}^*) - L(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\xi}}) > \tilde{\epsilon}\right) = 0$;

(ii) For any $\epsilon > 0$, we have $\lim_{n\to\infty} \Pr\left(D(\hat{\boldsymbol{\mu}}, \boldsymbol{\mu}^*) > \epsilon \text{ or } \hat{\boldsymbol{\xi}} \notin \Theta^*\right) = 0$.

Proof. The proof is provided in Section 2.7.4. □

Theorem 2.2. *Let $(\tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\xi}})$ be the maximizer of Eq. (2.7) with constraint $\|\boldsymbol{\xi}\|_0 \leq d_0$, and*

$$\nabla_n^+(d_0) = \max_{\|\boldsymbol{\xi}\|_0 > d_0} \frac{\widetilde{L}_n^{(\mathrm{SKKM})}(\tilde{\boldsymbol{\mu}}^{\boldsymbol{\xi}}, \boldsymbol{\xi}) - \widetilde{L}_n^{(\mathrm{SKKM})}(\tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\xi}})}{\|\boldsymbol{\xi}\|_0 - d_0}, \tag{2.16}$$

where $\tilde{\boldsymbol{\mu}}^{\boldsymbol{\xi}} = \arg\max_{\boldsymbol{\mu} \in \mathcal{H}_k^{\boldsymbol{\xi}}} \widetilde{L}_n^{(\mathrm{SKKM})}(\boldsymbol{\mu}, \boldsymbol{\xi})$. Under Conditions 2.1-2.4, and assume $\lim_{n\to\infty} \lambda_n = 0$ and $\lim_{n\to\infty} \Pr(\lambda_n > \nabla_n^+(d_0)) = 1$, then for any $\epsilon > 0$, we have

$$\lim_{n\to\infty} \Pr\left(D(\hat{\boldsymbol{\mu}}, \boldsymbol{\mu}^{**}) > \epsilon \text{ or } \hat{\boldsymbol{\xi}} \neq \boldsymbol{\xi}^{**}\right) = 0.$$

Proof. The proof is provided in Section 2.7.5. □

Corollary 2.1. *Under Conditions 2.1-2.4, and assume $\lim_{n\to\infty} \lambda_n = 0$ and $\lim_{n\to\infty} \lambda_n \sqrt{n} = \infty$, then for any $\epsilon > 0$, we have*

$$\lim_{n\to\infty} \Pr\left(D(\hat{\boldsymbol{\mu}}, \boldsymbol{\mu}^{**}) > \epsilon \text{ or } \hat{\boldsymbol{\xi}} \neq \boldsymbol{\xi}^{**}\right) = 0.$$

Proof. This is an immediate result of Theorem 2.2 and Lemma 2.6. □

We explain the above results briefly. First, Proposition 2.2 shows that in the population level, the optimal cluster centers must be $\boldsymbol{\mu}^{**}$, and the optimal indicator must belong to Θ^*, which implies that the optimizer of $L(\cdot, \cdot)$ could be multiple. Then, Theorem 2.1 provides the convergence in probability of $(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\xi}})$ to one of such optimizers. Moreover, under assumptions of λ_n, Theorem 2.2 guarantees that $(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\xi}})$ indeed converges in probability to the $(\boldsymbol{\mu}^{**}, \boldsymbol{\xi}^{**})$. Since $\boldsymbol{\mu}^{**}$ is the "true" cluster centers and $\boldsymbol{\xi}^{**}$ indicates the "true" relevant features, the convergence means the consistency of both clustering and feature selection.

We finally give some discussions on λ_n. First, Theorem 2.1 only requires that λ_n tends to 0, which guarantees that all relevant features can be selected. Second, Theorem 2.2 also requires that λ_n is slower than the order of $\nabla_n^+(d_0)$, which guarantees that all noise features would not be selected. Moreover, Corollary 2.1 shows that if λ_n tends to 0 slower than $O(1/\sqrt{n})$, then the consistency of feature selection can be guaranteed.

In addition, when h is the exponential kernel, since Proposition 2.2 shows that $(\boldsymbol{\mu}^{**}, \boldsymbol{\xi}^{**})$ is the unique maximizer of $L(\cdot, \cdot)$, then the requirement that λ_n tends to 0 is enough to guarantee the consistency of $(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\xi}})$ to $(\boldsymbol{\mu}^{**}, \boldsymbol{\xi}^{**})$, which implies the consistency of both clustering and feature selection.

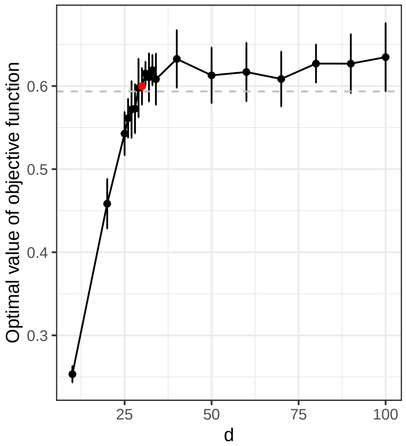


Figure 2.3: An example showing how the optimal value of objective function in Eq. (2.7) varies with the number of active features d. The x-axis is d and the y-axis is $\widetilde{Q}_n(d)$. The averaged $\widetilde{Q}_n(d)$ and one standard deviation of 10 repetitions are drawn. The red dot is the true value $d_0 = 30$. The gray dashed line is to express $\max_{s=1,\ldots,p} \widetilde{Q}_n(s) - \gamma_n$, and here is given by the averaged value of $\widetilde{Q}_n(p)$ minus its standard deviation.

Example 2.2. *For the constrained optimization problem Eq.(2.7), we introduce an empirical choice for d, that is, $\tilde{d}_n$ given in Eq.(2.12), which converges in probability to the true value d_0. We provide the formal proof in Section 2.7.7. Here, we explain the motivation of the empirical $\tilde{d}_n$ by an additional example showing how the optimal value of objective function $\widetilde{L}_n^{(\mathrm{SKKM})}$ varies with d. Suppose that the total number of features is $p = 100$ and the true number of relevant features is $d_0 = 30$. All $n = 1000$ data points x_i are independently drawn from the same Gaussian mixture distribution: $0.5\mathcal{N}(a, \mathrm{diag}(\sigma^2)) + 0.5\mathcal{N}(-a, \mathrm{diag}(\sigma^2))$, where $a_j \sim U[0.75, 1.25]$ if $j = 1, \ldots, d_0$, and $a_j = 0$ if $j = d_0 + 1, \ldots, p$, and $\sigma_j \sim U[0.75, 1.25]$ for all $j = 1, \ldots, p$. For each $d = 1, \ldots, p$, we run SKKM to obtain optimal value of objective function and denote it by $\widetilde{Q}_n(d)$. We calculate the average and standard deviation of 10 repetitions, and summarize the result in Figure 2.3, where the x-axis is d and the y-axis is $\widetilde{Q}_n(d)$. The red dot is the true value $d_0 = 30$. The gray dashed line is to express $\max_{s=1,\ldots,p} \widetilde{Q}_n(s) - \gamma_n$, and here is given by the averaged value of $\widetilde{Q}_n(p)$ minus its standard deviation. Hence, it can be seen that $\widetilde{Q}_n(d)$ increases when $d \leq d_0$, and becomes flat when $d \geq d_0$. The gray dashed line selects $\tilde{d}_n = 29$, which is very closed to $d_0 = 30$.*

2.6 Experiments

In this section, we evaluate the proposed method (SKKM) via numerical experiments on various datasets. The performance of the unweighted and weighted versions will be discussed separately. The specific structure of this section is as follows: (a) We first use a microarray dataset as an example to illustrate the advantage of the proposed method in Section 2.6.1. (b) Some benchmark datasets are used to compare the clustering performance of the proposed method and some existing methods in Section 2.6.2. (c) We further analyze the performance of feature selection on synthetic data in Section 2.6.3. (d) The consistency property is illustrated by simulations in Section 2.6.4. (e) The influence of tuning parameters is discussed based on the real data-based simulations in Section 2.6.5. (f) From Section 2.6.1 to Section 2.6.5, only the unweighted version of the proposed method (SKKM) is discussed. We evaluate the performance of the weighted version in Section 2.6.6.

Through this section, we use the Clustering Error Rate (CER) as the clustering performance index. Denote $\mathcal{C}^*$ to be the true partition of data points. The CER of the estimated partition $\widehat{\mathcal{C}}$ is defined as

$$\mathrm{CER}(\widehat{\mathcal{C}}, \mathcal{C}^*) = \frac{1}{\binom{n}{2}} \sum_{i>i'} \left| \mathbb{1}_{\widehat{\mathcal{C}}(i,i')} - \mathbb{1}_{\mathcal{C}^*(i,i')} \right|,$$

where $\mathbb{1}_{\mathcal{C}(i,i')} = 1$ if the i-th and i'-th data points are assigned to the same cluster according to the partition $\mathcal{C}$, 0 otherwise. Moreover, we consider sparse k-means (Witten & Tibshirani 2010), IF-PCA (Jin & Wang 2016) and Sparse MinMax k-means (Dey et al. 2020) as peer methods, and k-means and weighted kernel k-means as benchmark methods for comparisons.

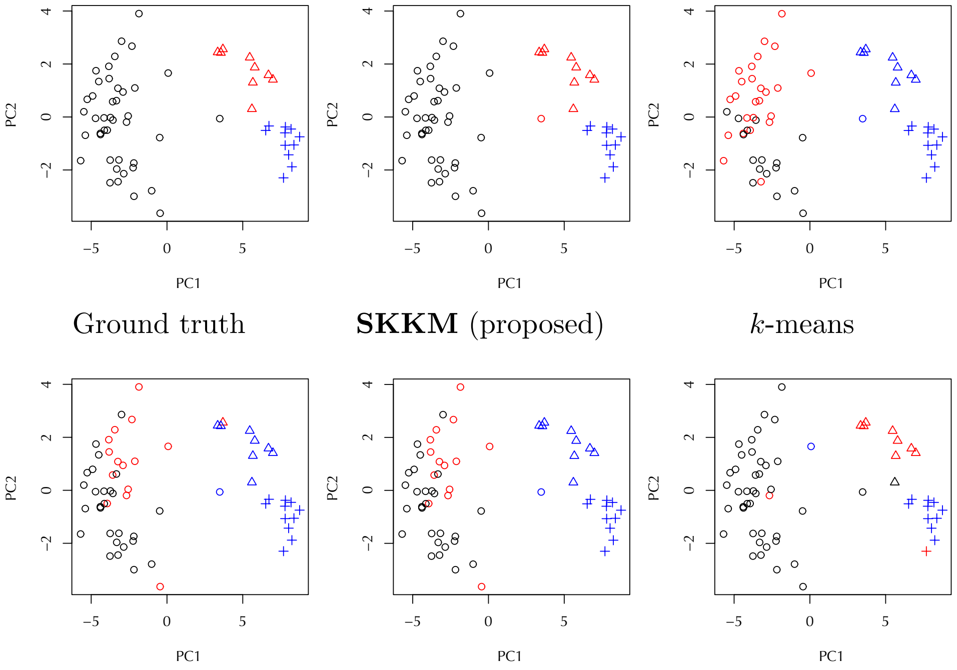
2.6.1 Case study

Microarray datasets are typical examples of high dimensional data where $p >> n$. They often consist of several thousand gene-expression levels but very few samples, which makes it difficult to analyze the underlying cluster structure (Jin & Wang 2016, Dey et al. 2020). We consider the *Lymphoma* dataset, which consists of 4026 gene expressions, collected over 62 samples. Out of the 62 samples, 42 are Diffuse Large B-Cell Lymphoma (DLBCL), 9 are Follicular Lymphoma (FL), and 11 are Chronic Lymphocytic Leukemia (CLL) cell samples. We use this dataset to illustrate the efficacy of the proposed sparse kernel k-means clustering (SKKM), in comparison with other peer algorithms. We run each algorithm 10 times and report the average

CERs in Table 2.1. It can be seen that the proposed method outperforms others. For visualization purposes, we performed the dimension reduction via PCA on the the feature selected by the proposed method, and illustrate the clustering results in Figure 2.4. It shows the superior clustering performance of the proposed method (SKKM). Figure 2.5 shows the feature weights and feature indicators against the corresponding features for each method. It can be easily seen that Sparse *k*-means clustering and Sparse MinMax *k*-means clustering do not assign zero weights to all the features, while the proposed method as well as IF-PCA assigns zero indicators to many of the features, and the proposed method (SKKM) derives the smallest set of selected features, which leads to the best performance.

Table 2.1: Average CERs on *Lymphoma* dataset

Dataset	IF-PCA	Sparse *k*-means	Sparse MinMax *k*-means	**SKKM**
Lymphoma	0.065	0.296	0.244	**0.026**



Ground truth

SKKM (proposed)

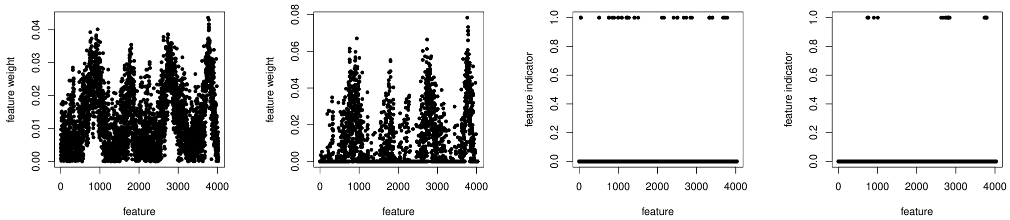
k-means

Sparse *k*-means

Sparse MinMax *k*-means

IF-PCA

Figure 2.4: The results of clustering on *Lymphoma* dataset of different methods. The x-axis and y-axis are the first two principle components of PCA.



Sparse k-means

Sparse MinMax k-means

IF-PCA

SKKM (proposed)

Figure 2.5: The results of selected features for the *Lymphoma* dataset of different methods. The x-axis is the index of each feature. The y-axis is the feature weight for Sparse k-means clustering and Sparse MinMax k-means clustering, while the feature indicator for IF-PCA and SKKM (proposed).

2.6.2 Comparison with existing methods via real-world data

In this section, we apply the proposed method (SKKM) to real-world datasets, and compare the performance of clustering with some existing methods for high-dimensional clustering, including IF-PCA (Jin & Wang 2016), Sparse k-means clustering (Witten & Tibshirani 2010) as well as one of its special application Sparse MinMax k-means clustering (Dey et al. 2020), classical kernel k-means clustering as well as k-means clustering. We consider UCI data and gene microarray data, both of which are benchmark sets for high-dimensional clustering[2].

The result of the comparison is summarized in Table 2.2. The reported values are averaged CERs of 10 repetitions. The numbers in parentheses represent the performance rankings of different clustering algorithms for a certain dataset. In the last row, we report the average rank of all algorithms. We also report the average numbers of selected features of different algorithms in Table 2.3. In the last row, we report the average ratio of selected features of all algorithms. It can be seen that in general, the proposed method (SKKM) achieves the comparably lowest classification error rates

[2] The first five UCI datasets are evaluated by Dey et al. (2020) as benchmark sets. The datasets *Glass*, *Breast*, *Vehicle* and *Control* can be found from `https://archive.ics.uci.edu/ml/index.php`. The *Trace* can be found from `http://www.cs.ucr.edu/~eamonn/time_series_data/`. The *Trace* and *Control* are time series datasets.
The last five high-dimensional gene microarray datasets are used by Jin & Wang (2016) and Dey et al. (2020), all of which can be found from `https://www.stat.cmu.edu/~jiashun/Research/software/GenomicsData/`.

(CERs) and thus obtains the highest average rank among the other sparse clustering algorithms, which illustrates the comparable performance of the proposed method to other peer methods.

Table 2.2: Comparison of CERs of different algorithms for the real-world datasets

Dataset	k	$n \times p$	k-means	Kernel k-means	IF-PCA	Sparse k-means	Sparse MinMax k-means	**SKKM** (proposed)
Glass	6	214×9	0.328 (1)	0.346 (4)	0.351 (5)	0.328 (1)	0.374 (6)	0.335 (3)
Breast	2	699×9	0.080 (2)	0.087 (5)	0.133 (6)	0.082 (3)	0.047 (1)	0.085 (4)
Vehicle	4	846×18	0.402 (5)	0.387 (4)	0.382 (3)	0.348 (1)	0.548 (6)	0.349 (2)
Trace	4	200×275	0.251 (4)	0.250 (1)	0.495 (6)	0.250 (1)	0.445 (5)	0.250 (1)
Control	6	600×60	0.156 (2)	0.168 (3)	0.295 (6)	0.147 (1)	0.200 (4)	0.168 (3)
Brain	5	42×5597	0.375 (6)	0.241 (5)	0.119 (1)	0.190 (3)	0.238 (4)	0.175 (2)
Colon	2	62×2000	0.508 (6)	0.505 (3)	0.403 (2)	0.506 (5)	0.145 (1)	0.505 (3)
Leukemia	2	72×3571	0.394 (3)	0.412 (6)	0.069 (2)	0.407 (5)	0.028 (1)	0.394 (3)
Lymphoma	3	62×4026	0.297 (5)	0.029 (2)	0.065 (3)	0.297 (5)	0.244 (4)	0.026 (1)
SRBCT	4	63×2308	0.371 (5)	0.388 (6)	0.318 (1)	0.358 (3)	0.333 (2)	0.359 (4)
Avg. rank			3.9	3.9	3.5	2.8	3.4	**2.6**

Table 2.3: Comparison of numbers of selected features for the real-world datasets

Dataset	p	IF-PCA	Sparse k-means	Sparse MinMax k-means	**SKKM** (proposed)
Glass	9	4	9	9	7
Breast	9	5	9	9	8
Vehicle	18	4	18	18	14
Trace	275	105	275	275	165
Control	60	26	60	60	60
Brain	5597	429	5597	5597	17
Colon	2000	25	2000	663	28
Leukemia	3571	213	3571	765	37
Lymphoma	4026	44	4026	1432	39
SRBCT	2308	54	2308	716	21
Avg. ratio		22%	100%	72%	41%

2.6.3 Discussion on feature selection

In this section, we analyze the effect of feature selection of the proposed method (SKKM) on synthetic datasets. These synthetic datasets consist of low-dimensional ground truth relevant features and many noise features. The following types of noise features are considered: (a) Independent normal noise feature. (b) Correlated normal noise feature. The covariance matrix is set to be $(\rho^{|s-t|/3})_{s,t=1}^{p-d}$, where ρ follows a uniform distribution, i.e., $\rho \sim U[0.1, 0.9]$. (c) Independent $\chi^2(5)$ noise feature. The details of all synthetic datasets are provided in Section A.2 of Appendix A. Since the ground truth relevant feature is already known for synthetic datasets, we use F1score of whether relevant features are selected or not as the criterion. For sparse k-means clustering and sparse MinMax k-means clustering, the jth feature is selected if its weight $w_j > 0$. For the sake of space economy, we denote the sparse k-means clustering by *SKM* and denote the sparse MinMax k-means clustering by *SMMKM* for short notations.

The results are summarized in Table 2.4. The reported values are the averaged F1scores of 20 repetitions, as well as corresponding standard deviations. In addition, we also report the corresponding precision and recall values. It can be seen that for the proposed sparse kernel k-means clustering outperforms in identifying all the relevant features and thus leads to higher Precision, Recall and F1score values. However, the sparse k-means clustering identifies only a subset of the relevant features to be important.

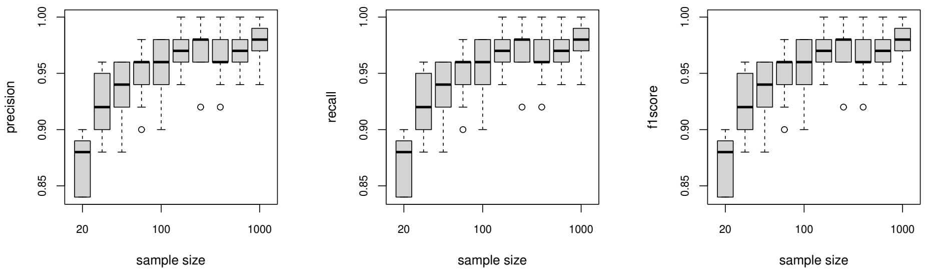
2.6.4 Consistency analysis

In this section, we show the consistency of feature selection as the sample size $n \to \infty$ through some numerical simulation. In this section, we consider the simple Gaussian mixture distribution with $k = 2$ classes, and the number of all features and the number of relevant features are fixed to be $p = 200$ and $d = 50$, respectively. Specifically, the simulation data matrix $\mathrm{X}_{n\times p}$ is generated in the following way. For the i-th observation, the cluster (C_1 or C_2) to which it belongs is specified by the Bernoulli distribution with expectation equal to 0.5, and the element x_{ij} is sampled from $\mathcal{N}(\mu_{ij}, \sigma_j^2)$. The μ_{ij} is specified as follow: (1) for any $i \in C_1$ $j \leq d$, we take μ_{ij} from the uniform distribution $U[0.75, 1.25]$; (2) for any $i \in C_2$ $j \leq d$, we take μ_{ij} from the uniform distribution $U[-1.25, -0.75]$; (3) for all i and any $j > d$, we take $\mu_{ij} = 0$.

Figure 2.6 illustrates the trend of F1scores (as well as Precision and Recall) of whether relevant features are selected or not as the sample size n increases. With each sample size, we report the average value of 20 repetitions. Figure 2.7 shows the frequency of each feature being selected within 20 simulations. To save space, we take three relevant features and three noise features as examples. It can be seen that as $n \to \infty$, F1score (as well as Precision and Recall) of feature selection tends to 1, which means that the estimator $\hat{\boldsymbol{\xi}}$ is consistent. Moreover, the frequency of being selected of relevant features is close to 1, while that of noise features is close 0 as $n \to \infty$, which means that the relevant features could be almost recognized as $n \to \infty$.

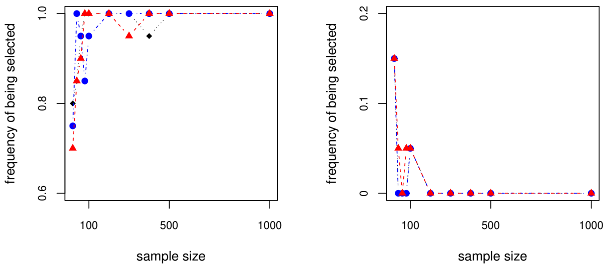
Table 2.4: Comparison of F1scores of different algorithms for synthetic datasets

Dataset	Precision				Recall				F1score			
	IF-PCA	SKM	SMMKM	**SKKM**	IF-PCA	SKM	SMMKM	**SKKM**	IF-PCA	SKM	SMMKM	**SKKM**
data1	0.499 (0.17)	0.231 (0.00)	0.231 (0.00)	0.950 (0.12)	0.700 (0.10)	1.000 (0.00)	1.000 (0.00)	0.950 (0.12)	0.562 (0.12)	0.375 (0.00)	0.375 (0.00)	**0.950** **(0.12)**
data2	0.566 (0.16)	0.217 (0.04)	0.231 (0.00)	0.617 (0.35)	0.700 (0.10)	0.930 (0.18)	1.000 (0.00)	0.617 (0.35)	0.562 (0.12)	0.351 (0.06)	0.375 (0.00)	**0.617** **(0.35)**
data3	0.050 (0.01)	0.290 (0.00)	0.034 (0.00)	0.617 (0.35)	0.850 (0.17)	1.000 (0.00)	1.000 (0.00)	0.617 (0.35)	0.094 (0.02)	0.057 (0.00)	0.065 (0.00)	**0.617** **(0.35)**
data4	0.051 (0.03)	0.029 (0.00)	0.036 (0.00)	0.700 (0.21)	0.633 (0.28)	1.000 (0.00)	1.000 (0.00)	0.700 (0.21)	0.092 (0.05)	0.057 (0.00)	0.069 (0.01)	**0.700** **(0.21)**
data5	0.440 (0.13)	0.167 (0.00)	0.167 (0.00)	1.000 (0.00)	1.000 (0.00)	1.000 (0.00)	1.000 (0.00)	1.000 (0.00)	0.601 (0.12)	0.286 (0.00)	0.286 (0.00)	**1.000** **(0.00)**
data6	0.453 (0.13)	0.148 (0.01)	0.158 (0.02)	0.525 (0.30)	1.000 (0.00)	0.800 (0.30)	0.925 (0.18)	0.525 (0.30)	**0.596** **(0.12)**	0.263 (0.07)	0.270 (0.04)	0.525 (0.30)
data7	0.128 (0.11)	0.020 (0.00)	0.033 (0.00)	0.825 (0.24)	1.000 (0.00)	1.000 (0.00)	1.000 (0.00)	0.825 (0.24)	0.212 (0.16)	0.040 (0.00)	0.064 (0.01)	**0.825** **(0.24)**
data8	0.009 (0.01)	0.022 (0.00)	0.044 (0.01)	0.921 (0.19)	0.175 (0.24)	1.000 (0.00)	1.000 (0.00)	0.921 (0.19)	0.050 (0.01)	0.043 (0.01)	0.083 (0.02)	**0.921** **(0.19)**



Precision Recall F1score

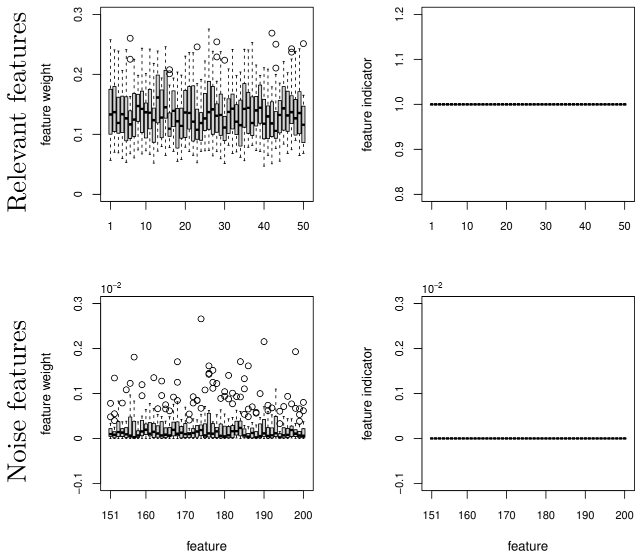
Figure 2.6: The results of consistency of feature selection.



Relevant features Noise features

Figure 2.7: The results of the consistency of feature selection. The x-axis is the sample size, and the y-axis is the frequency of being selected over 20 repetitions. The left panel illustrates the trend of three relevant features. The right panel illustrates the trend of three noise features.

Furthermore, we compare the performance of feature selection with Sparse k-means clustering under large sample size ($n = 1000$). We show the estimators of the feature indicator ξ_j of our method over 20 repetitions in Figure 2.8, where all 50 relevant features and 50 noise features are illustrated in the top and bottom rows, respectively. For comparison, we also show the estimators of feature weight w_j of sparse k-means clustering in the left column. It can be seen that the proposed method assigns exact zero to noise features and exact one to relevant features, while Sparse k-means clustering has a larger variability in assigning weights to relevant features and leaves many noise features to be selected.



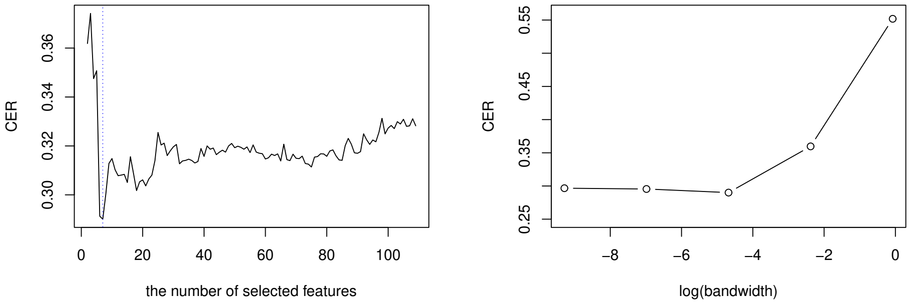
Sparse k-means **SKKM** (proposed)

Figure 2.8: The results of feature weights w_j assigned by sparse k-means clustering and feature indicators ξ_j assigned by SKKM (proposed) with large sample size ($n = 1000$). The x-axis represents the index of feature, and the y-axis represents the box plots of w_j and ξ_j. The top row illustrates the result of relevant features. The bottom row illustrates the result of noise features.

2.6.5 Sensitivity of tuning parameters

In this section, we analyze the influence of tuning parameters, including the number of selected features and the bandwidth of the kernel function. For the choice of the bandwidth, although the median heuristic is applied, it is still important to investigate how different bandwidths affect the performance of the proposed method. We consider the bandwidth ν_0^2 derived by median heuristic (Eq. (2.6)), as well as $\{0.01\nu_0^2, 0.1\nu_0^2, 10\nu_0^2, 100\nu_0^2\}$. Through this section, we consider the simulation based on real-world data. The benchmark UCI dataset *Glass* is used. Since it contains nine features, we artificially add 100 white noise features to get the full data matrix with $n = 214$ and $p = 109$.

Figure 2.9 illustrates the trend of CERs as the number of selected features and the bandwidth vary, respectively. The reported lines are the averaged values of 10 repetitions. It can be seen that when a small set of features (seven features suggested by gap statistics in this example) are selected, the clustering result is the best. On one hand, when the size of selected features is excessively small, due to the lack of information, the error of clustering would increase. On the other hand, when more irrelevant features are selected, the clustering performance would also be reduced. Moreover, when the bandwidth ν^2 of kernel function is excessively small, the clustering performance of the proposed method would be significantly reduced, while a larger bandwidth ν^2 is a relatively safe choice.



Number of selected features

Bandwidth

Figure 2.9: The influence of the tuning parameters. The left panel is about the number of selected features. The right panel is about the inverse of bandwidth $\frac{1}{\nu^2}$ (logarithm) of the kernel function.

2.6.6 Performance of the weighted version

In this section, we discuss the performance of the weighted version of the proposed method. To distinguish with unweighted version, through this section we denote the weighted version of the proposed method by SWKKM. To specify the weights w_i of observations, we consider the specific technology used in weighted kernel k-means clustering that coincides with the normalized cut. More details are provided in the Section A.3.1 of Appendix A.

Table 2.5: Averaged CERs of different methods on *ORL* dataset.

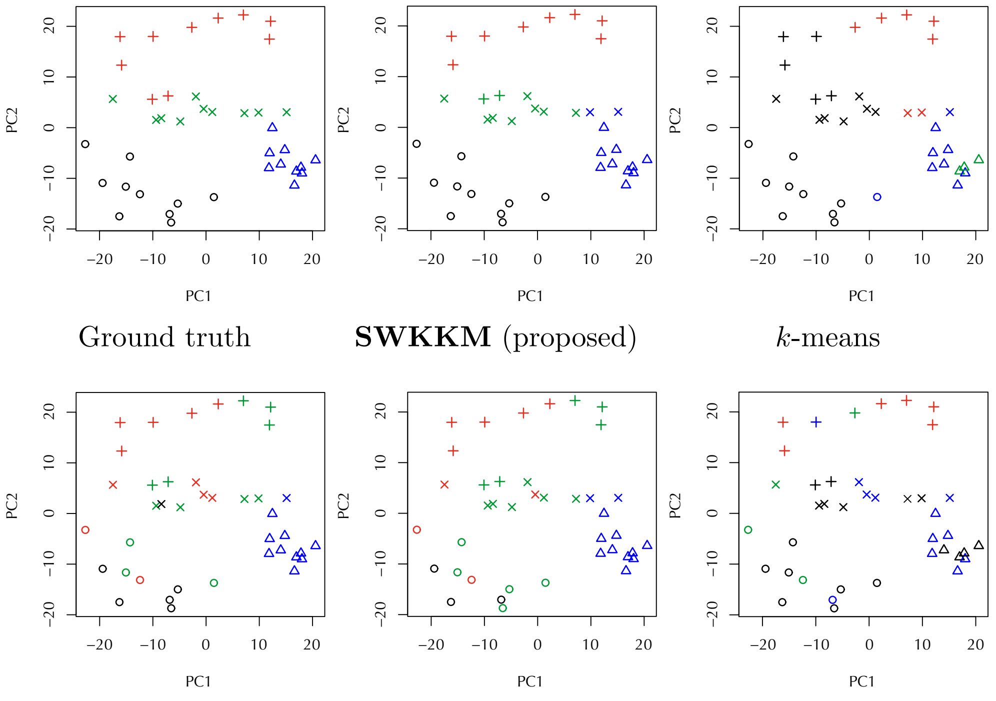
Dataset	k-means	Weighted kernel k-means	IF-PCA	Sparse k-means	**SWKKM**
ORL	0.304	0.231	0.253	0.242	**0.075**

We similarly give an example to illustrate the performance of the proposed method (SWKKM). This case is a real-world dataset *ORL*, which is a face image dataset. The *ORL* data consists of 40 images ($n = 40$) and 1024 pixel values ($p = 1024$). Each of 4 distinct subjects ($k = 4$) has ten images that were taken at different times, varying the lighting, facial expressions and facial details. We run each algorithm 10 times and report the average value of CERs in Table 2.5, and similarly use PCA as the visualization technique to illustrate the clustering result of each algorithm in Figure 2.10. It can be seen that the proposed method (SWKKM) has lower CER and thus outperforms other peer algorithms in clustering.

We apply the proposed method (SWKKM) on real-world datasets to compare with peer methods. All ten datasets are from an open-source feature selection repository called *scikit-feature*[3]. We report the average CERs of each algorithm in Table 2.6. It can be seen that the proposed method (SWKKM) has lower CERs than other methods on more than half of real-world datasets, and thus gets the highest rank, which confirms the better performance of the proposed method.

More experiments results of evaluating the performance of the weighted version (SWKKM) on synthetic datasets are provided in Section A.3.2 of Appendix A.

[3] The *scikit-feature* repository is an open-source feature selection repository in Python developed at Arizona State University: `https://jundongl.github.io/scikit-feature/datasets.html`.



Ground truth SWKKM (proposed) *k*-means

Sparse *k*-means Weighted kernel *k*-means IF-PCA

Figure 2.10: The results of clustering on the *ORL* dataset of different methods. The x-axis and y-axis are the first two principle components on PCA.

Table 2.6: Comparison of CERs of different algorithms for the real-world datasets.

Dataset	k	$n \times p$	k-means	Weighted kernel k-means	IF-PCA	Sparse k-means	**SWKKM** (proposed)
Lung discrete	7	73×325	0.149 (5)	0.140 (4)	0.136 (3)	0.083 (1)	0.121 (2)
GLIOMA	4	50×4434	0.275 (4)	0.276 (5)	0.174 (1)	0.269 (3)	0.255 (2)
Lsolet	5	300×617	0.195 (4)	0.160 (3)	0.207 (5)	0.150 (2)	0.125 (1)
Leukemia	2	72×7070	0.437 (5)	0.409 (3)	0.306 (2)	0.419 (4)	0.243 (1)
Lung	4	197×3312	0.241 (5)	0.104 (2)	0.179 (4)	0.128 (3)	0.069 (1)
Lymphoma	4	76×4026	0.264 (3)	0.296 (5)	0.270 (4)	0.221 (2)	0.197 (1)
ORL	4	40×1024	0.304 (5)	0.231 (2)	0.253 (4)	0.242 (3)	0.075 (1)
TOX171	4	171×5748	0.350 (3)	0.350 (4)	0.239 (1)	0.380 (5)	0.293 (2)
WarpAR10P	10	130×2400	0.192 (5)	0.186 (4)	0.093 (1)	0.180 (3)	0.171 (2)
Yale	15	165×1024	0.119 (5)	0.117 (4)	0.068 (1)	0.099 (3)	0.085 (2)
Avg. Rank			4.4	3.6	2.6	2.9	**1.5**

2.7 Proofs

2.7.1 Preliminaries

We provide more details of exponential kernel. Through the theoretical analysis, we focus on the form $h(x, \tilde{x}) = \exp(\langle x, \tilde{x} \rangle_2)$ for any $x, \tilde{x} \in \mathcal{X}$, where $\langle x, \tilde{x} \rangle_2 = \sum_{j=1}^{p} x_j \tilde{x}_j$, and x_j and $\tilde{x}_j$ are j-th component of x and $\tilde{x}$, respectively. According to Lemma 4.8 of Steinwart & Christmann (2008), by expanding the exponential function $\exp(\cdot)$ into its Taylor series at 0, we write the kernel function as

$$\begin{aligned} h(x, \tilde{x}) = \exp\left(\sum_{j=1}^{p} x_j \tilde{x}_j \right) &= \sum_{t=0}^{\infty} \frac{1}{t!} \left(\sum_{j=1}^{p} x_j \tilde{x}_j \right)^t \\ &= \sum_{s_1, \dots, s_p \geq 0} \frac{1}{\prod_{j=1}^{p} s_j!} \prod_{j=1}^{p} x_j^{s_j} \tilde{x}_j^{s_j}. \end{aligned}$$

Let $\boldsymbol{s} = (s_1, \dots, s_p) \in \mathbb{N}_0^p$ be a multiple index and $\mathbb{N}_0$ be the set of non-negative integers. The RKHS $\mathcal{H}$ is given by

$$\mathcal{H} = \left\{ g(x) = \sum_{\boldsymbol{s}} c_{\boldsymbol{s}} \prod_{j=1}^{p} x_j^{s_j} \,\middle|\, \sum_{\boldsymbol{s}} c_{\boldsymbol{s}}^2 \prod_{j=1}^{p} s_j! < \infty \right\},$$

where $c_{\boldsymbol{s}}$ is a constant for a given index $\boldsymbol{s}$ in $\mathbb{N}_0^p$. The inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ is given by

$$\langle g, \tilde{g} \rangle_{\mathcal{H}} = \sum_{\boldsymbol{s}} c_{\boldsymbol{s}} \cdot \tilde{c}_{\boldsymbol{s}} \cdot \prod_{j=1}^{p} s_j!$$

for any $g(x) = \sum\limits_{\boldsymbol{s}} c_{\boldsymbol{s}} \prod_{j=1}^{p} x_j^{s_j}$ and $\tilde{g}(x) = \sum\limits_{\boldsymbol{s}} \tilde{c}_{\boldsymbol{s}} \prod_{j=1}^{p} x_j^{s_j}$. The norm $\| \cdot \|_{\mathcal{H}}$ is given by $\|g\|_{\mathcal{H}} = \sqrt{\langle g, g \rangle_{\mathcal{H}}}$. The kernel h has the reproducing property that $\langle g, h(\cdot, x) \rangle_{\mathcal{H}} = g(x)$ for any $g \in \mathcal{H}$ and $x \in \mathcal{X}$.

Moreover, the mapping $\psi : \mathcal{X} \to \mathcal{H}$ defined by $\psi(x) = h(\cdot, x)$ has the expression as follows, and thus $h(x, \tilde{x}) = \langle \psi(\tilde{x}), \psi(x) \rangle_{\mathcal{H}}$.

$$\psi(x) = \left(\frac{1}{\sqrt{\prod_{j=1}^{p} s_j!}} \prod_{j=1}^{p} x_j^{s_j} \right)_{s_1, \dots, s_p \geq 0}. \tag{2.17}$$

Furthermore, for any $\boldsymbol{\eta} \in \{0, 1\}^p$, the sparse counterpart of $\mathcal{H}$ is given by

$$\mathcal{H}^{\boldsymbol{\eta}} = \left\{ g(x) = \sum_{\boldsymbol{s}} c_{\boldsymbol{s}} \prod_{j:\eta_j=1} x_j^{s_j} \prod_{j:\eta_j=0} \mathbb{1}(s_j = 0) \,\middle|\, \sum_{\boldsymbol{s}} c_{\boldsymbol{s}}^2 \prod_{j:\eta_j=1} s_j! \prod_{j:\eta_j=0} \mathbb{1}(s_j = 0) < \infty \right\}.$$

As introduced in Section 2.5, the exponential kernel satisfies Assumption 2.1.

Lemma 2.1. *Suppose that h is the exponential kernel with the form $h(x, \tilde{x}) = \exp(\langle x, \tilde{x}\rangle_2)$, and the associated RKHS of it is $\mathcal{H}$. Let $\psi : \mathcal{X} \to \mathcal{H}$ be $\psi(x) = h(\cdot, x)$. For any $\boldsymbol{\eta}, \tilde{\boldsymbol{\eta}} \in \{0, 1\}^p$, if the support of $\boldsymbol{\eta}$ is included in the support of $\tilde{\boldsymbol{\eta}}$, that is, $\{j = 1, \dots, p \mid \eta_j = 1\} \subset \{j = 1, \dots, p \mid \tilde{\eta}_j = 1\}$, then we have*

(i) $\mathcal{H}^{\boldsymbol{\eta}} \subset \mathcal{H}^{\tilde{\boldsymbol{\eta}}}$;

(ii) $\langle g, \psi^{\boldsymbol{\eta}}(x)\rangle_{\mathcal{H}} = \langle g, \psi^{\tilde{\boldsymbol{\eta}}}(x)\rangle_{\mathcal{H}} = g(x)$, for any $g \in \mathcal{H}^{\boldsymbol{\eta}}, x \in \mathcal{X}$.

Proof. (i) For any $g \in \mathcal{H}^{\boldsymbol{\eta}}$, there exists a sequence of constants $c_{\boldsymbol{s}}$ indexed by $\boldsymbol{s}$ and satisfying

$$\sum_{\boldsymbol{s}} c_{\boldsymbol{s}}^2 \prod_{j:\eta_j=1} s_j! \prod_{j:\eta_j=0} \mathbb{1}(s_j = 0) < \infty,$$

such that $g : \mathcal{X} \to \mathbb{R}$ has the form of

$$g(x) = \sum_{\boldsymbol{s}} c_{\boldsymbol{s}} \prod_{j:\eta_j=1} x_j^{s_j} \prod_{j:\eta_j=0} \mathbb{1}(s_j = 0).$$

For any index $\boldsymbol{s}$, if we take $\tilde{c}_{\boldsymbol{s}} = c_{\boldsymbol{s}} \prod_{j:\tilde{\eta}_j=1,\eta_j=0} \mathbb{1}(s_j = 0)$, then we have

$$\begin{aligned}
&\sum_{\boldsymbol{s}} \tilde{c}_{\boldsymbol{s}}^2 \prod_{j:\tilde{\eta}_j=1} s_j! \prod_{j:\tilde{\eta}_j=0} \mathbb{1}(s_j = 0) \\
&= \sum_{\boldsymbol{s}} c_{\boldsymbol{s}}^2 \prod_{j:\tilde{\eta}_j=1,\eta_j=0} \mathbb{1}(s_j = 0) \prod_{j:\tilde{\eta}_j=1} s_j! \prod_{j:\tilde{\eta}_j=0} \mathbb{1}(s_j = 0) \\
&= \sum_{\boldsymbol{s}} c_{\boldsymbol{s}}^2 \prod_{j:\tilde{\eta}_j=1,\eta_j=1} s_j! \prod_{j:\eta_j=0} \mathbb{1}(s_j = 0) \\
&= \sum_{\boldsymbol{s}} c_{\boldsymbol{s}}^2 \prod_{j:\eta_j=1} s_j! \prod_{j:\eta_j=0} \mathbb{1}(s_j = 0) < \infty,
\end{aligned}$$

where the last equality is because $\{j : \eta_j = 1\} \subset \{j : \tilde{\eta}_j = 1\}$. Therefore, the function $\tilde{g} : \mathcal{X} \to \mathbb{R}$ given by

$$\tilde{g}(x) = \sum_{\boldsymbol{s}} \tilde{c}_{\boldsymbol{s}} \prod_{j:\tilde{\eta}_j=1} x_j^{s_j} \prod_{j:\tilde{\eta}_j=0} \mathbb{1}(s_j = 0)$$

is an element of $\mathcal{H}^{\tilde{\boldsymbol{\eta}}}$. Moreover, since

$$\begin{aligned}\tilde{g}(x) &= \sum_{\boldsymbol{s}} c_{\boldsymbol{s}} \prod_{j:\tilde{\eta}_j=1,\eta_j=0} \mathbb{1}(s_j=0) \prod_{j:\tilde{\eta}_j=1} x_j^{s_j} \prod_{j:\tilde{\eta}_j=0} \mathbb{1}(s_j=0) \\ &= \sum_{\boldsymbol{s}} c_{\boldsymbol{s}} \prod_{j:\tilde{\eta}_j=1,\eta_j=1} x_j^{s_j} \prod_{j:\eta_j=0} \mathbb{1}(s_j=0) \\ &= \sum_{\boldsymbol{s}} c_{\boldsymbol{s}} \prod_{j:\eta_j=1} x_j^{s_j} \prod_{j:\eta_j=0} \mathbb{1}(s_j=0) = g(x),\end{aligned}$$

we have g is an element of $\mathcal{H}^{\tilde{\boldsymbol{\eta}}}$.

(ii) For any $g \in \mathcal{H}^{\boldsymbol{\eta}}$, according to the definition of $\psi^{\boldsymbol{\eta}}(x) = h^{\boldsymbol{\eta}}(\cdot, x)$ and the reproducing property, we have $\langle g, \psi^{\boldsymbol{\eta}}(x)\rangle_{\mathcal{H}} = g(x)$. Similarly, since g is also an element of $\mathcal{H}^{\tilde{\boldsymbol{\eta}}}$, it still holds that $\langle g, \psi^{\tilde{\boldsymbol{\eta}}}(x)\rangle_{\mathcal{H}} = g(x)$. □

Next, we briefly explain how Assumption 2.1 facilitates our analysis.

Specifically, let us consider any arbitrary reproducing kernel $h : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ and its RKHS $\mathcal{H}$, which is equipped with an inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ and a norm given by $\|g\|_{\mathcal{H}} = \sqrt{\langle g, g\rangle_{\mathcal{H}}}$. The mapping $\psi : \mathcal{X} \to \mathcal{H}$ is defined by $\psi(x) = h(\cdot, x)$ for any $x \in \mathcal{X}$. Then, for any $\boldsymbol{\xi} \in \{0,1\}^p$, we can construct a new kernel $h^{\boldsymbol{\xi}} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ by $h^{\boldsymbol{\xi}}(x, \tilde{x}) = h(x \circ \boldsymbol{\xi}, \tilde{x} \circ \boldsymbol{\xi})$. The corresponding RKHS is denoted by $\mathcal{H}^{\boldsymbol{\xi}}$, which is equipped with the inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}^{\boldsymbol{\xi}}}$ and a norm given by $\|g\|_{\mathcal{H}^{\boldsymbol{\xi}}} = \sqrt{\langle g, g\rangle_{\mathcal{H}^{\boldsymbol{\xi}}}}$. We define a mapping $\psi^{\boldsymbol{\xi}} : \mathcal{X} \to \mathcal{H}^{\boldsymbol{\xi}}$ by $\psi^{\boldsymbol{\xi}}(x) = h(\cdot \circ \boldsymbol{\xi}, x \circ \boldsymbol{\xi})$ for any $x \in \mathcal{X}$.

To analyze the consistency of SKKM, we also consider the reformulation Eq. (2.7) about cluster centers and feature indicator. For any arbitrary reproducing kernel h, a more accurate version of Eq. (2.7) is

$$\max_{\substack{\boldsymbol{\xi} \in \{0,1\}^p \\ \mu_l^{\boldsymbol{\xi}} \in \mathcal{H}^{\boldsymbol{\xi}}, \forall l=1,\ldots,k}} \frac{1}{n} \sum_{i=1}^{n} \left\{ \|\psi^{\boldsymbol{\xi}}(X_i) - \hat{\mu}_0^{\boldsymbol{\xi}}\|_{\mathcal{H}^{\boldsymbol{\xi}}}^2 - \min_{l=1,\ldots,k} \|\psi^{\boldsymbol{\xi}}(X_i) - \mu_l^{\boldsymbol{\xi}}\|_{\mathcal{H}^{\boldsymbol{\xi}}}^2 \right\} \quad \text{s.t. } \|\boldsymbol{\xi}\|_0 \le d,$$

where we use the notation $\mu_l^{\boldsymbol{\xi}}$ to express the l-th cluster center only for showing the connection to Eq.(2.4). The $\mu_l^{\boldsymbol{\xi}} \in \mathcal{H}^{\boldsymbol{\xi}}$ means that the search region depends on $\boldsymbol{\xi}$. Hence, to find the optimal solution, we need to consider $\bigcup_{\boldsymbol{\xi}} \mathcal{H}^{\boldsymbol{\xi}}$, while $\mathcal{H}^{\boldsymbol{\xi}} \subset \mathcal{H}$ does not necessarily hold[4] for all $\boldsymbol{\xi} \in \{0,1\}^p$. This makes the analysis more complicated.

Fortunately, if h satisfies Assumption 2.1 (E.g.: h is the exponential kernel), then we have $\mathcal{H}^{\boldsymbol{\xi}} \subset \mathcal{H}$ hold for all $\boldsymbol{\xi} \in \{0,1\}^p$. It allows us to use the inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ and norm $\| \cdot \|_{\mathcal{H}}$ on any $\mathcal{H}^{\boldsymbol{\xi}}$, which leads to the reformulation Eq. (2.7) as well as Eq. (2.9).

[4] For example, if $h(x, \tilde{x}) = \exp(-\|x - \tilde{x}\|_2^2)$ is a Gaussian kernel, then for $\boldsymbol{\xi} = \mathbf{0}_p$, we have for any $\tilde{x} \in \mathcal{X}$, the function $h(\cdot \circ \mathbf{0}_p, \tilde{x} \circ \mathbf{0}_p) : \mathcal{X} \to \mathbb{R}$ is a constant function in $\mathcal{H}^{\mathbf{0}}$, while it is not included in $\mathcal{H}$. It means that $\mathcal{H}^{\mathbf{0}}$ is not a subset of $\mathcal{H}$.

2.7.2 Proof of Proposition 2.1

In this section, we prove Proposition 2.1 about the equivalence between the optimization problems Eq. (2.7) and Eq. (2.9).

Recall that Eq. (2.7) is a constrained problem with $\|\boldsymbol{\xi}\|_0 \leq d$, where $d = 1, \ldots, p$ is the tuning parameter, and the maximizer is denoted by $(\tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\xi}})$. Eq. (2.9) is an unconstrained problem with a penalty term $\lambda_n \|\boldsymbol{\xi}\|_0$, where $\lambda_n > 0$ is the tuning parameter, and the maximizer is denoted by $(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\xi}})$. Therefore, it suffices to prove: (i) For any λ_n, there exists a d such that $(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\xi}})$ is a maximizer of Eq. (2.7); (ii) For any d, there exists a λ_n such that $(\tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\xi}})$ is a maximizer of Eq. (2.9).

(i) For any λ_n, since $(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\xi}})$ maximizes $\widehat{L}_n(\boldsymbol{\mu}, \boldsymbol{\xi}) = \widetilde{L}_n^{(\mathrm{SKKM})}(\boldsymbol{\mu}, \boldsymbol{\xi}) - \lambda_n \|\boldsymbol{\xi}\|_0$, we take $d_{\lambda_n} = \|\hat{\boldsymbol{\xi}}\|_0$. Then, for any $(\boldsymbol{\mu}, \boldsymbol{\xi})$ satisfying $\|\boldsymbol{\xi}\|_0 \leq d_{\lambda_n}$, we have

$$\begin{aligned} \widetilde{L}_n^{(\mathrm{SKKM})}(\boldsymbol{\mu}, \boldsymbol{\xi}) - \lambda_n \|\boldsymbol{\xi}\|_0 &\leq \widetilde{L}_n^{(\mathrm{SKKM})}(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\xi}}) - \lambda_n \|\hat{\boldsymbol{\xi}}\|_0 \\ \text{and} \quad \lambda_n \|\boldsymbol{\xi}\|_0 &\leq \lambda_n \|\hat{\boldsymbol{\xi}}\|_0. \end{aligned}$$

It follows that $\widetilde{L}_n^{(\mathrm{SKKM})}(\boldsymbol{\mu}, \boldsymbol{\xi}) \leq \widetilde{L}_n^{(\mathrm{SKKM})}(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\xi}})$, which means that $(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\xi}})$ is also a maximizer of Eq. (2.7) with $d = d_{\lambda_n}$.

(ii) For any d, the $(\tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\xi}})$ maximizes $\widetilde{L}_n^{(\mathrm{SKKM})}(\boldsymbol{\mu}, \boldsymbol{\xi})$ with the constraint $\|\boldsymbol{\xi}\|_0 \leq d$. Here, we suppose $\|\tilde{\boldsymbol{\xi}}\|_0 = d$, otherwise, we can instead consider maximizing $\widetilde{L}_n^{(\mathrm{SKKM})}(\boldsymbol{\mu}, \boldsymbol{\xi})$ with the constraint $\|\boldsymbol{\xi}\|_0 \leq \|\tilde{\boldsymbol{\xi}}\|_0$. If there exists λ_n satisfying Eq. (2.11), then $\widehat{L}_n(\boldsymbol{\mu}, \boldsymbol{\xi})$ with such λ_n is maximized by $(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\xi}})$.

If $(\tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\xi}})$ is not a maximizer of $\widehat{L}_n(\boldsymbol{\mu}, \boldsymbol{\xi})$ with such λ_n, there must be

$$\widetilde{L}_n^{(\mathrm{SKKM})}(\tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\xi}}) - \lambda_n \|\tilde{\boldsymbol{\xi}}\|_0 < \widetilde{L}_n^{(\mathrm{SKKM})}(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\xi}}) - \lambda_n \|\hat{\boldsymbol{\xi}}\|_0.$$

When $\|\hat{\boldsymbol{\xi}}\|_0 = d$, it means that $\widetilde{L}_n^{(\mathrm{SKKM})}(\tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\xi}}) < \widetilde{L}_n^{(\mathrm{SKKM})}(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\xi}})$, which is a contradiction. When $\|\hat{\boldsymbol{\xi}}\|_0 < d$, it means that

$$\frac{\widetilde{L}_n^{(\mathrm{SKKM})}(\tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\xi}}) - \widetilde{L}_n^{(\mathrm{SKKM})}(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\xi}})}{d - \|\hat{\boldsymbol{\xi}}\|_0} < \lambda_n,$$

which contradicts to upper bound of Eq. (2.11). When $\|\hat{\boldsymbol{\xi}}\|_0 > d$, it means that

$$\frac{\widetilde{L}_n^{(\mathrm{SKKM})}(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\xi}}) - \widetilde{L}_n^{(\mathrm{SKKM})}(\tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\xi}})}{\|\hat{\boldsymbol{\xi}}\|_0 - d} > \lambda_n,$$

which contradicts to lower bound of Eq. (2.11). Therefore, $(\tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\xi}})$ is indeed a maximizer of $\widehat{L}_n(\boldsymbol{\mu}, \boldsymbol{\xi})$ with λ_n satisfying Eq. (2.11).

We complete the proof.

2.7.3 Proof of Proposition 2.2

In this section, we prove Proposition 2.2 about the optimal solution $(\boldsymbol{\mu}^*, \boldsymbol{\xi}^*)$ of $L(\boldsymbol{\mu}, \boldsymbol{\xi})$ under settings and conditions given in Section 2.5.

Recall that the maximizer of $L(\boldsymbol{\mu}, \mathbf{1}_p)$ denoted by $\boldsymbol{\mu}^{**} = \{\mu_1^{**}, \ldots, \mu_k^{**}\}$ (Eq.(2.15)) can be viewed as the optimal cluster centers given by kernel k-means using all features. Each μ_l^{**} only relies on $x_1, \ldots, x_{d_0}$ under Condition 2.1. Since $\mu_l^{**} \in \mathcal{H}$, if h is the exponential kernel, then we can write $\mu_l^{**} : \mathcal{X} \to \mathbb{R}$ to be

$$\mu_l^{**}(x) = \sum_{\boldsymbol{s}} c_{l,\boldsymbol{s}}^{**} \cdot \prod_{j=1}^{d_0} x_j^{s_j} \cdot \prod_{j=d_0+1}^{p} \mathbb{1}(s_j = 0),$$

where $c_{l,\boldsymbol{s}}^{**}$ is a given constant associated with μ_l^{**} and indexed by $\boldsymbol{s}$. Moreover, recall that $\Theta^* = \{\boldsymbol{\xi} \in \{0,1\}^p \mid \xi_j = 1, \forall j = 1, \ldots, d_0\}$ and $\boldsymbol{\xi}^{**} = (\mathbf{1}_{d_0}, \mathbf{0}_{p-d_0})$. It also follows that $\mu_l^{**} \in \mathcal{H}^{\boldsymbol{\xi}^{**}}$ for any $l = 1, \ldots, k$.

Our first aim is to prove $\boldsymbol{\mu}^* = \boldsymbol{\mu}^{**}$ and $\boldsymbol{\xi}^* \in \Theta^*$. Since the search region for maximizer of $L(\boldsymbol{\mu}, \boldsymbol{\xi})$ is $\{(\boldsymbol{\mu}, \boldsymbol{\xi}) \mid \boldsymbol{\xi} \in \{0,1\}^p, \mu_l \in \mathcal{H}^{\boldsymbol{\xi}}, \forall l = 1, \ldots, k\}$, it can be divided into two parts by $\boldsymbol{\xi}$ belonging to Θ^* or not. Under Condition 2.4, we know that the maximal value of $L(\boldsymbol{\mu}, \boldsymbol{\xi})$ would not be obtained if $\boldsymbol{\xi} \notin \Theta^*$. It follows that the maximizer $(\boldsymbol{\mu}^*, \boldsymbol{\xi}^*)$ must satisfy $\boldsymbol{\xi}^* \in \Theta^*$. Therefore, it suffices to prove: Given $\boldsymbol{\xi}^* \in \Theta^*$, $L(\boldsymbol{\mu}, \boldsymbol{\xi}^*)$ is only maximized by $\boldsymbol{\mu} = \boldsymbol{\mu}^{**}$, which is Lemma 2.2.

Our second aim is to prove $(\boldsymbol{\mu}^{**}, \boldsymbol{\xi}^{**})$ is one of maximizers of $L(\boldsymbol{\mu}, \boldsymbol{\xi})$. Based on Lemma 2.2, it suffices to prove: Given $\boldsymbol{\mu}^* = \boldsymbol{\mu}^{**}$, $L(\boldsymbol{\mu}^*, \boldsymbol{\xi})$ can be maximized by $\boldsymbol{\xi} = \boldsymbol{\xi}^{**}$, which is Lemma 2.3.

Our third aim is to prove that when h is the exponential kernel, $(\boldsymbol{\mu}^{**}, \boldsymbol{\xi}^{**})$ is the unique maximizer of $L(\boldsymbol{\mu}, \boldsymbol{\xi})$, which is an immediate result of Lemma 2.2 and the second part of Lemma 2.3.

Lemma 2.2. *Given $\boldsymbol{\xi}^* \in \Theta^*$, we have*

$$\boldsymbol{\mu}^{**} = \underset{\mu_l \in \mathcal{H}^{\boldsymbol{\xi}^*}, \forall l = 1, \ldots, k}{\arg\max} L(\boldsymbol{\mu}, \boldsymbol{\xi}^*).$$

Proof. At first, according to the definition of $\boldsymbol{\mu}^{**}$, the result holds for $\boldsymbol{\xi}^* = \mathbf{1}_p$. It suffices to consider $\boldsymbol{\xi}^* \in \Theta^*$ with the form of $(\mathbf{1}_{d'}, \mathbf{0}_{p-d'})$ for some fixed $d' \in \{d_0, d_0 + 1, \ldots, p - 1\}$.

Since the search region relies on $\boldsymbol{\xi}^*$, we should find $\boldsymbol{\mu} = \{\mu_1, \ldots, \mu_k\}$ that maximizes $L(\boldsymbol{\mu}, \boldsymbol{\xi}^*)$ with restricting each $\mu_l \in \mathcal{H}^{\boldsymbol{\xi}^*}$. Here we note that because the support of $\boldsymbol{\xi}^{**}$ is included in the support of $\boldsymbol{\xi}^*$, then Assumption 2.1 gives $\mathcal{H}^{\boldsymbol{\xi}^{**}} \subset \mathcal{H}^{\boldsymbol{\xi}^*} \subset \mathcal{H}$.

According to Eq.(2.13), the $L(\boldsymbol{\mu}, \boldsymbol{\xi}^*)$ is given by

$$L(\boldsymbol{\mu}, \boldsymbol{\xi}^*) = \int_{\mathcal{X}} \|\psi^{\boldsymbol{\xi}^*}(x) - \mu_0^{\boldsymbol{\xi}^*}\|_{\mathcal{H}}^2 \, d\mathbb{P}(x) - \int_{\mathcal{X}} \min_{l=1,\dots,k} \|\psi^{\boldsymbol{\xi}^*}(x) - \mu_l\|_{\mathcal{H}}^2 \, d\mathbb{P}(x),$$

where the first term is a constant unrelated to $\boldsymbol{\mu}$. Thus, maximizing $L(\boldsymbol{\mu}, \boldsymbol{\xi}^*)$ is equivalent to minimizing $\int_{\mathcal{X}} \min_{l=1,\dots,k} \|\psi^{\boldsymbol{\xi}^*}(x) - \mu_l\|_{\mathcal{H}}^2 \, d\mathbb{P}(x)$. It suffices to prove

$$\boldsymbol{\mu}^{**} = \operatorname*{arg\,min}_{\mu_1,\dots,\mu_k \in \mathcal{H}^{\boldsymbol{\xi}^*}} \int_{\mathcal{X}} \min_{l=1,\dots,k} \|\psi^{\boldsymbol{\xi}^*}(x) - \mu_l\|_{\mathcal{H}}^2 \, d\mathbb{P}(x).$$

If $\boldsymbol{\mu}^{**}$ is not the solution, then there exists a different $\boldsymbol{\nu} = \{\nu_1, \dots, \nu_k\}$ with each $\nu_l \in \mathcal{H}^{\boldsymbol{\xi}^*}$ such that

$$\int_{\mathcal{X}} \min_{l=1,\dots,k} \|\psi^{\boldsymbol{\xi}^*}(x) - \nu_l\|_{\mathcal{H}}^2 \, d\mathbb{P}(x) < \int_{\mathcal{X}} \min_{l=1,\dots,k} \|\psi^{\boldsymbol{\xi}^*}(x) - \mu_l^{**}\|_{\mathcal{H}}^2 \, d\mathbb{P}(x). \tag{2.18}$$

On the other hand, $\boldsymbol{\mu}^{**}$ uniquely maximizes $L(\boldsymbol{\mu}, \mathbf{1}_p)$, where

$$L(\boldsymbol{\mu}, \mathbf{1}_p) = \int_{\mathcal{X}} \|\psi(x) - \mu_0\|_{\mathcal{H}}^2 \, d\mathbb{P}(x) - \int_{\mathcal{X}} \min_{l=1,\dots,k} \|\psi(x) - \mu_l\|_{\mathcal{H}}^2 \, d\mathbb{P}(x).$$

Since the first term of $L(\boldsymbol{\mu}, \mathbf{1}_p)$ is a constant, it implies that $\boldsymbol{\mu}^{**}$ uniquely minimizes $\int_{\mathcal{X}} \min_{l=1,\dots,k} \|\psi(x) - \mu_l\|_{\mathcal{H}}^2 \, d\mathbb{P}(x)$. Then, we have

$$\int_{\mathcal{X}} \min_{l=1,\dots,k} \|\psi(x) - \mu_l^{**}\|_{\mathcal{H}}^2 \, d\mathbb{P}(x) < \int_{\mathcal{X}} \min_{l=1,\dots,k} \|\psi(x) - \nu_l\|_{\mathcal{H}}^2 \, d\mathbb{P}(x). \tag{2.19}$$

Combining Eq.(2.18) and Eq.(2.19), we have

$$\begin{aligned} &\int_{\mathcal{X}} \min_{l=1,\dots,k} \|\psi(x) - \mu_l^{**}\|_{\mathcal{H}}^2 - \min_{l=1,\dots,k} \|\psi^{\boldsymbol{\xi}^*}(x) - \mu_l^{**}\|_{\mathcal{H}}^2 \, d\mathbb{P}(x) \\ &< \int_{\mathcal{X}} \min_{l=1,\dots,k} \|\psi(x) - \nu_l\|_{\mathcal{H}}^2 - \min_{l=1,\dots,k} \|\psi^{\boldsymbol{\xi}^*}(x) - \nu_l\|_{\mathcal{H}}^2 \, d\mathbb{P}(x). \end{aligned} \tag{2.20}$$

For the left hand of Eq.(2.20), for any $x \in \mathcal{X}$, we denote $l_x = \operatorname*{arg\,min}_{l=1,\dots,k} \|\psi(x) - \mu_l^{**}\|_{\mathcal{H}}^2$, then we have

$$\begin{aligned} &\min_{l=1,\dots,k} \|\psi(x) - \mu_l^{**}\|_{\mathcal{H}}^2 - \min_{l=1,\dots,k} \|\psi^{\boldsymbol{\xi}^*}(x) - \mu_l^{**}\|_{\mathcal{H}}^2 \\ &= \|\psi(x) - \mu_{l_x}^{**}\|_{\mathcal{H}}^2 - \min_{l=1,\dots,k} \|\psi^{\boldsymbol{\xi}^*}(x) - \mu_l^{**}\|_{\mathcal{H}}^2 \\ &\geq \|\psi(x) - \mu_{l_x}^{**}\|_{\mathcal{H}}^2 - \|\psi^{\boldsymbol{\xi}^*}(x) - \mu_{l_x}^{**}\|_{\mathcal{H}}^2 \\ &= \Big[\langle \psi(x), \psi(x)\rangle_{\mathcal{H}} - \langle \psi^{\boldsymbol{\xi}^*}(x), \psi^{\boldsymbol{\xi}^*}(x)\rangle_{\mathcal{H}}\Big] - 2 \cdot \Big[\langle \mu_{l_x}^{**}, \psi(x)\rangle_{\mathcal{H}} - \langle \mu_{l_x}^{**}, \psi^{\boldsymbol{\xi}^*}(x)\rangle_{\mathcal{H}}\Big] \\ &= \Big[h(x,x) - h(x \circ \boldsymbol{\xi}^*, x \circ \boldsymbol{\xi}^*)\Big] - 2 \cdot \Big[\mu_{l_x}^{**}(x) - \mu_{l_x}^{**}(x)\Big] \\ &= h(x,x) - h(x \circ \boldsymbol{\xi}^*, x \circ \boldsymbol{\xi}^*). \end{aligned}$$

The third equality is because $\mu^{**}_{l_x} \in \mathcal{H}^{\boldsymbol{\xi}^{**}} \subset \mathcal{H}^{\boldsymbol{\xi}^{*}} \subset \mathcal{H}$, then according to Assumption 2.1 we get the result.

For the right hand Eq.(2.20), for any $x \in \mathcal{X}$, we denote $l'_x = \arg\min_{l=1,\dots,k} \|\psi^{\boldsymbol{\xi}^*}(x) - \nu_l\|^2_{\mathcal{H}}$, then we have

$$
\begin{aligned}
&\min_{l=1,\dots,k} \|\psi(x) - \nu_l\|^2_{\mathcal{H}} - \min_{l=1,\dots,k} \|\psi^{\boldsymbol{\xi}^*}(x) - \nu_l\|^2_{\mathcal{H}} \\
&= \min_{l=1,\dots,k} \|\psi(x) - \nu_l\|^2_{\mathcal{H}} - \|\psi^{\boldsymbol{\xi}^*}(x) - \nu_{l'_x}\|^2_{\mathcal{H}} \\
&\leq \|\psi(x) - \nu_{l'_x}\|^2_{\mathcal{H}} - \|\psi^{\boldsymbol{\xi}^*}(x) - \nu_{l'_x}\|^2_{\mathcal{H}} \\
&= \left[\langle \psi(x), \psi(x)\rangle_{\mathcal{H}} - \langle \psi^{\boldsymbol{\xi}^*}(x), \psi^{\boldsymbol{\xi}^*}(x)\rangle_{\mathcal{H}}\right] - 2 \cdot \left[\langle \nu_{l'_x}, \psi(x)\rangle_{\mathcal{H}} - \langle \nu_{l'_x}, \psi^{\boldsymbol{\xi}^*}(x)\rangle_{\mathcal{H}}\right] \\
&= \left[h(x,x) - h(x \circ \boldsymbol{\xi}^*, x \circ \boldsymbol{\xi}^*)\right] - 2 \cdot \left[\nu_{l'_x}(x) - \nu_{l'_x}(x)\right] \\
&= h(x,x) - h(x \circ \boldsymbol{\xi}^*, x \circ \boldsymbol{\xi}^*),
\end{aligned}
$$

where the third equality is because $\nu_{l'_x} \in \mathcal{H}^{\boldsymbol{\xi}^*} \subset \mathcal{H}$.

Combining two bounds leads to $h(x,x) - h(x \circ \boldsymbol{\xi}^*, x \circ \boldsymbol{\xi}^*) < h(x,x) - h(x \circ \boldsymbol{\xi}^*, x \circ \boldsymbol{\xi}^*)$ for any $x \in \mathcal{X}$, which is a contradiction. We complete the proof. □

Lemma 2.3. *Given $\boldsymbol{\mu}^* = \boldsymbol{\mu}^{**}$, we have for each $\boldsymbol{\xi} \in \Theta^*$,*

$$L(\boldsymbol{\mu}^*, \boldsymbol{\xi}^{**}) \geq L(\boldsymbol{\mu}^*, \boldsymbol{\xi}).$$

In addition, when h is the exponential kernel, then $L(\boldsymbol{\mu}^*, \boldsymbol{\xi}^{**}) > L(\boldsymbol{\mu}^*, \boldsymbol{\xi})$.

Proof. We only consider $\boldsymbol{\zeta}$ with the form of $(\mathbf{1}_{d'}, \mathbf{0}_{p-d'})$ for some fixed $d' \in \{d_0 + 1, \dots, p\}$, which can be easily extended to other cases.

First, according to the definition of L, we have

$$
\begin{aligned}
&L(\boldsymbol{\mu}^*, \boldsymbol{\xi}^{**}) - L(\boldsymbol{\mu}^*, \boldsymbol{\zeta}) \\
&= \left[\int_{\mathcal{X}} \|\psi^{\boldsymbol{\xi}^{**}}(x) - \mu_0^{\boldsymbol{\xi}^{**}}\|^2_{\mathcal{H}} - \min_{l=1,\dots,k} \|\psi^{\boldsymbol{\xi}^{**}}(x) - \mu^*_l\|^2_{\mathcal{H}}\, d\mathbb{P}(x)\right] \\
&\quad - \left[\int_{\mathcal{X}} \|\psi^{\boldsymbol{\zeta}}(x) - \mu_0^{\boldsymbol{\zeta}}\|^2_{\mathcal{H}} - \min_{l=1,\dots,k} \|\psi^{\boldsymbol{\zeta}}(x) - \mu^*_l\|^2_{\mathcal{H}}\, d\mathbb{P}(x)\right] \\
&= \underbrace{\int_{\mathcal{X}} \|\psi^{\boldsymbol{\xi}^{**}}(x) - \mu_0^{\boldsymbol{\xi}^{**}}\|^2_{\mathcal{H}} - \|\psi^{\boldsymbol{\zeta}}(x) - \mu_0^{\boldsymbol{\zeta}}\|^2_{\mathcal{H}}\, d\mathbb{P}(x)}_{\text{(I)}} \\
&\quad - \underbrace{\int_{\mathcal{X}} \min_{l=1,\dots,k} \|\psi^{\boldsymbol{\xi}^{**}}(x) - \mu^*_l\|^2_{\mathcal{H}} - \min_{l=1,\dots,k} \|\psi^{\boldsymbol{\zeta}}(x) - \mu^*_l\|^2_{\mathcal{H}}\, d\mathbb{P}(x)}_{\text{(II)}}.
\end{aligned}
$$

For (I), according to the definitions of $\boldsymbol{\mu}_0^{\boldsymbol{\xi}^{**}}$ and $\boldsymbol{\mu}_0^{\boldsymbol{\zeta}}$, we have

$$
\begin{aligned}
(\mathrm{I}) &= \int_{\mathcal{X}} \langle \psi^{\boldsymbol{\xi}^{**}}(x), \psi^{\boldsymbol{\xi}^{**}}(x)\rangle_{\mathcal{H}} - \langle \psi^{\boldsymbol{\zeta}}(x), \psi^{\boldsymbol{\zeta}}(x)\rangle_{\mathcal{H}} \; d\mathbb{P}(x) \\
&\quad - 2 \cdot \int_{\mathcal{X}} \langle \mu_0^{\boldsymbol{\xi}^{**}}, \psi^{\boldsymbol{\xi}^{**}}(x)\rangle_{\mathcal{H}} - \langle \mu_0^{\boldsymbol{\zeta}}, \psi^{\boldsymbol{\zeta}}(x)\rangle_{\mathcal{H}} \; d\mathbb{P}(x) \\
&\quad + \int_{\mathcal{X}} \langle \mu_0^{\boldsymbol{\xi}^{**}}, \mu_0^{\boldsymbol{\xi}^{**}}\rangle_{\mathcal{H}} - \langle \mu_0^{\boldsymbol{\zeta}}, \mu_0^{\boldsymbol{\zeta}}\rangle_{\mathcal{H}} \; d\mathbb{P}(x) \\
&= \int_{\mathcal{X}} h(x \circ \boldsymbol{\xi}^{**}, x \circ \boldsymbol{\xi}^{**}) - h(x \circ \boldsymbol{\zeta}, x \circ \boldsymbol{\zeta}) \; d\mathbb{P}(x) \\
&\quad - 2 \cdot \left(\|\mu_0^{\boldsymbol{\xi}^{**}}\|_{\mathcal{H}}^2 - \|\mu_0^{\boldsymbol{\zeta}}\|_{\mathcal{H}}^2 \right) + \left(\|\mu_0^{\boldsymbol{\xi}^{**}}\|_{\mathcal{H}}^2 - \|\mu_0^{\boldsymbol{\zeta}}\|_{\mathcal{H}}^2 \right) \\
&= \mathbb{E}_{X \sim \mathbb{P}} \Big[h(X \circ \boldsymbol{\xi}^{**}, X \circ \boldsymbol{\xi}^{**}) - h(X \circ \boldsymbol{\zeta}, X \circ \boldsymbol{\zeta}) \Big] \\
&\quad - \mathbb{E}_{X \sim \mathbb{P}} \mathbb{E}_{\tilde{X} \sim \mathbb{P}} \Big[h(X \circ \boldsymbol{\xi}^{**}, \tilde{X} \circ \boldsymbol{\xi}^{**}) - h(X \circ \boldsymbol{\zeta}, \tilde{X} \circ \boldsymbol{\zeta}) \Big].
\end{aligned}
$$

For (II), since for all $l = 1, \dots, k$, $\mu_l^* \in \mathcal{H}^{\boldsymbol{\xi}^{**}} \subset \mathcal{H}^{\boldsymbol{\zeta}}$, then for any $x \in \mathcal{X}$, we have $\langle \mu_l^*, \psi^{\boldsymbol{\xi}^{**}}(x)\rangle_{\mathcal{H}} = \mu_l^*(x)$ and $\langle \mu_l^*, \psi^{\boldsymbol{\zeta}}(x)\rangle_{\mathcal{H}} = \mu_l^*(x)$. For a fixed $x \in \mathcal{X}$, if we denote $l_x = \underset{l=1,\dots,k}{\arg\min} \|\psi^{\boldsymbol{\xi}^{**}}(x) - \mu_l^*\|_{\mathcal{H}}^2$, then

$$
\begin{aligned}
&\min_{l=1,\dots,k} \|\psi^{\boldsymbol{\xi}^{**}}(x) - \mu_l^*\|_{\mathcal{H}}^2 - \min_{l=1,\dots,k} \|\psi^{\boldsymbol{\zeta}}(x) - \mu_l^*\|_{\mathcal{H}}^2 \\
&= \|\psi^{\boldsymbol{\xi}^{**}}(x) - \mu_{l_x}^*\|_{\mathcal{H}}^2 - \min_{l=1,\dots,k} \|\psi^{\boldsymbol{\zeta}}(x) - \mu_l^*\|_{\mathcal{H}}^2 \\
&\geq \|\psi^{\boldsymbol{\xi}^{**}}(x) - \mu_{l_x}^*\|_{\mathcal{H}}^2 - \|\psi^{\boldsymbol{\zeta}}(x) - \mu_{l_x}^*\|_{\mathcal{H}}^2 \\
&= \Big[\langle \psi^{\boldsymbol{\xi}^{**}}(x), \psi^{\boldsymbol{\xi}^{**}}(x)\rangle_{\mathcal{H}} - \langle \psi^{\boldsymbol{\zeta}}(x), \psi^{\boldsymbol{\zeta}}(x)\rangle_{\mathcal{H}} \Big] - 2 \cdot \Big[\langle \mu_{l_x}^*, \psi^{\boldsymbol{\xi}^{**}}(x)\rangle_{\mathcal{H}} - \langle \mu_{l_x}^*, \psi^{\boldsymbol{\zeta}}(x)\rangle_{\mathcal{H}} \Big] \\
&= \Big[h(x \circ \boldsymbol{\xi}^{**}, x \circ \boldsymbol{\xi}^{**}) - h(x \circ \boldsymbol{\zeta}, x \circ \boldsymbol{\zeta}) \Big] - 2 \cdot \Big[\mu_{l_x}^*(x) - \mu_{l_x}^*(x) \Big] \\
&= h(x \circ \boldsymbol{\xi}^{**}, x \circ \boldsymbol{\xi}^{**}) - h(x \circ \boldsymbol{\zeta}, x \circ \boldsymbol{\zeta}).
\end{aligned}
$$

Similarly, if we denote $l'_x = \underset{l=1,\ldots,k}{\arg\min} \|\psi^{\boldsymbol{\zeta}}(x) - \mu_l^*\|_{\mathcal{H}}^2$, then

$$\begin{aligned}
&\min_{l=1,\ldots,k} \|\psi^{\boldsymbol{\xi}^{**}}(x) - \mu_l^*\|_{\mathcal{H}}^2 - \min_{l=1,\ldots,k} \|\psi^{\boldsymbol{\zeta}}(x) - \mu_l^*\|_{\mathcal{H}}^2 \\
&= \min_{l=1,\ldots,k} \|\psi^{\boldsymbol{\xi}^{**}}(x) - \mu_l^*\|_{\mathcal{H}}^2 - \|\psi^{\boldsymbol{\zeta}}(x) - \mu_{l'_x}^*\|_{\mathcal{H}}^2 \\
&\leq \|\psi^{\boldsymbol{\xi}^{**}}(x) - \mu_{l'_x}^*\|_{\mathcal{H}}^2 - \|\psi^{\boldsymbol{\zeta}}(x) - \mu_{l'_x}^*\|_{\mathcal{H}}^2 \\
&= \Big[\langle \psi^{\boldsymbol{\xi}^{**}}(x), \psi^{\boldsymbol{\xi}^{**}}(x)\rangle_{\mathcal{H}} - \langle \psi^{\boldsymbol{\zeta}}(x), \psi^{\boldsymbol{\zeta}}(x)\rangle_{\mathcal{H}}\Big] - 2 \cdot \Big[\langle \mu_{l'_x}^*, \psi^{\boldsymbol{\xi}^{**}}(x)\rangle_{\mathcal{H}} - \langle \mu_{l'_x}^*, \psi^{\boldsymbol{\zeta}}(x)\rangle_{\mathcal{H}}\Big] \\
&= \Big[h(x \circ \boldsymbol{\xi}^{**}, x \circ \boldsymbol{\xi}^{**}) - h(x \circ \boldsymbol{\zeta}, x \circ \boldsymbol{\zeta})\Big] - 2 \cdot \Big[\mu_{l'_x}^*(x) - \mu_{l'_x}^*(x)\Big] \\
&= h(x \circ \boldsymbol{\xi}^{**}, x \circ \boldsymbol{\xi}^{**}) - h(x \circ \boldsymbol{\zeta}, x \circ \boldsymbol{\zeta}).
\end{aligned}$$

It follows that

$$\begin{aligned}
\text{(II)} &= \int_{\mathcal{X}} h(x \circ \boldsymbol{\xi}^{**}, x \circ \boldsymbol{\xi}^{**}) - h(x \circ \boldsymbol{\zeta}, x \circ \boldsymbol{\zeta}) \; d\mathbb{P}(x) \\
&= \mathbb{E}_{X\sim\mathbb{P}}\Big[h(X \circ \boldsymbol{\xi}^{**}, X \circ \boldsymbol{\xi}^{**}) - h(X \circ \boldsymbol{\zeta}, X \circ \boldsymbol{\zeta})\Big].
\end{aligned}$$

Therefore, combining (I) and (II) leads to

$$\begin{aligned}
L(\boldsymbol{\mu}^*, \boldsymbol{\xi}^{**}) - L(\boldsymbol{\mu}^*, \boldsymbol{\zeta}) &= \text{(I)} - \text{(II)} \\
&= -\mathbb{E}_{X\sim\mathbb{P}}\mathbb{E}_{\tilde{X}\sim\mathbb{P}}\Big[h(X \circ \boldsymbol{\xi}^{**}, \tilde{X} \circ \boldsymbol{\xi}^{**}) - h(X \circ \boldsymbol{\zeta}, \tilde{X} \circ \boldsymbol{\zeta})\Big].
\end{aligned}$$

Finally, under Condition 2.3, we obtain $L(\boldsymbol{\mu}^*, \boldsymbol{\xi}^{**}) - L(\boldsymbol{\mu}^*, \boldsymbol{\zeta}) \geq 0$. We complete the proof of the first part.

Next, when h is exponential kernel, we have

$$\begin{aligned}
\mathbb{E}_{X\sim\mathbb{P}}\mathbb{E}_{\tilde{X}\sim\mathbb{P}}\big[h(X \circ \boldsymbol{\xi}^{**}, \tilde{X} \circ \boldsymbol{\xi}^{**})\big] &= \int_{\mathcal{X}}\int_{\mathcal{X}} h(x \circ \boldsymbol{\xi}^{**}, \tilde{x} \circ \boldsymbol{\xi}^{**}) \; d\mathbb{P}(\tilde{x}) \; d\mathbb{P}(x) \\
&= \int_{\mathcal{X}}\int_{\mathcal{X}} \exp\left(\sum_{j=1}^{d_0} x_j \tilde{x}_j\right) \; d\mathbb{P}(\tilde{x}) \; d\mathbb{P}(x)
\end{aligned}$$

and

$$\begin{aligned}
\mathbb{E}_{X\sim\mathbb{P}}\mathbb{E}_{\tilde{X}\sim\mathbb{P}}\big[h(X \circ \boldsymbol{\zeta}, \tilde{X} \circ \boldsymbol{\zeta})\big] &= \int_{\mathcal{X}}\int_{\mathcal{X}} h(x \circ \boldsymbol{\zeta}, \tilde{x} \circ \boldsymbol{\zeta}) \; d\mathbb{P}(\tilde{x}) \; d\mathbb{P}(x) \\
&= \int_{\mathcal{X}}\int_{\mathcal{X}} \exp\left(\sum_{j=1}^{d'} x_j \tilde{x}_j\right) \; d\mathbb{P}(\tilde{x}) \; d\mathbb{P}(x) \\
&= \int_{\mathcal{X}}\int_{\mathcal{X}} \exp\left(\sum_{j=1}^{d_0} x_j \tilde{x}_j\right) \cdot \prod_{j=d_0+1}^{d'} \exp(x_j \tilde{x}_j) \; d\mathbb{P}(\tilde{x}) \; d\mathbb{P}(x).
\end{aligned}$$

Using Jensen's inequality, we have for any $j = d_0 + 1, \dots, d'$,

$$\begin{aligned}\mathbb{E}_{X_j \sim \mathbb{P}_j} \mathbb{E}_{\tilde{X}_j \sim \mathbb{P}_j} \big[\exp(X_j \tilde{X}_j) \big] &\geq \exp \left(\mathbb{E}_{X_j \sim \mathbb{P}_j} \mathbb{E}_{\tilde{X}_j \sim \mathbb{P}_j} \left[X_j \tilde{X}_j \right] \right) \\ &= \exp \left(\left(\mathbb{E}_{X_j \sim \mathbb{P}_j} [X_j] \right)^2 \right) \geq 1,\end{aligned}$$

where the equality holds only if $\mathbb{P}_j$ is a degenerated distribution. Thus, by using Condition 2.2, we have the following holds for any non-degenerated distribution:

$$\mathbb{E}_{X \sim \mathbb{P}} \mathbb{E}_{\tilde{X} \sim \mathbb{P}} \big[h(X \circ \boldsymbol{\xi}^{**}, \tilde{X} \circ \boldsymbol{\xi}^{**}) \big] < \mathbb{E}_{X \sim \mathbb{P}} \mathbb{E}_{\tilde{X} \sim \mathbb{P}} \big[h(X \circ \boldsymbol{\zeta}, \tilde{X} \circ \boldsymbol{\zeta}) \big].$$

which implies that $L(\boldsymbol{\mu}^*, \boldsymbol{\xi}^{**}) - L(\boldsymbol{\mu}^*, \boldsymbol{\zeta}) > 0$. We complete the proof. □

2.7.4 Proof of Theorem 2.1

In this section, we prove Theorem 2.1 about the convergence in probability of $(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\xi}})$ under settings and conditions given in Section 2.5.

(i) Recall that $\boldsymbol{\xi}^{**} = (\mathbf{1}_{d_0}, \mathbf{0}_{p-d_0}) \in \Theta^*$ and $\mu_l^{**} \in \mathcal{H}^{\boldsymbol{\xi}^{**}}$ for any $l = 1, \dots, k$. According to Proposition 2.2 and the definition of $(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\xi}})$, we have

$$\begin{aligned}
&L(\boldsymbol{\mu}^*, \boldsymbol{\xi}^*) - L(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\xi}}) \\
&= L(\boldsymbol{\mu}^{**}, \boldsymbol{\xi}^{**}) - L(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\xi}}) \\
&= L(\boldsymbol{\mu}^{**}, \boldsymbol{\xi}^{**}) - \widehat{L}_n(\boldsymbol{\mu}^{**}, \boldsymbol{\xi}^{**}) + \widehat{L}_n(\boldsymbol{\mu}^{**}, \boldsymbol{\xi}^{**}) - \widehat{L}_n(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\xi}}) + \widehat{L}_n(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\xi}}) - L(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\xi}}) \\
&\leq 2 \cdot \sup_{\substack{\boldsymbol{\xi} \in \{0,1\}^p \\ \mu_l \in \mathcal{H}^{\boldsymbol{\xi}}, \forall l = 1, \dots, k}} \left| \widehat{L}_n(\boldsymbol{\mu}, \boldsymbol{\xi}) - L(\boldsymbol{\mu}, \boldsymbol{\xi}) \right| .
\end{aligned}$$

By using Lemma 2.4, for any $\tilde{\epsilon} > 0$, there exists $N_1 \in \mathbb{N}_+$ such that for any $n \geq N_1$, it holds that

$$\Pr\left(L(\boldsymbol{\mu}^*, \boldsymbol{\xi}^*) - L(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\xi}}) \leq \tilde{\epsilon} \right) \geq 1 - 4 \exp\left(- \left[\frac{(\tilde{\epsilon}/2 - p\lambda_n)\sqrt{n} - C_1}{C_2} \right]^2 \right),$$

where C_1 and C_2 are constants given by Eq. (2.25). Moreover, since $\lim_{n \to \infty} \lambda_n = 0$, then for any $\tilde{\delta} > 0$ satisfying $\tilde{\epsilon}/2 - p\tilde{\delta} > 0$, there exists $N_2 \in \mathbb{N}$ such that $\lambda_n < \tilde{\delta}$ for $n \geq N_2$. It follows that for any $n \geq \max\{N_1, N_2\}$, we have

$$\Pr\left(L(\boldsymbol{\mu}^*, \boldsymbol{\xi}^*) - L(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\xi}}) \leq \tilde{\epsilon} \right) \geq 1 - 4 \exp\left(- \left[\frac{(\tilde{\epsilon}/2 - p\tilde{\delta})\sqrt{n} - C_1}{C_2} \right]^2 \right).$$

Therefore, $\lim_{n \to \infty} \Pr\left(L(\boldsymbol{\mu}^*, \boldsymbol{\xi}^*) - L(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\xi}}) > \tilde{\epsilon} \right) = 0$.

(ii) First, for a fixed optimal indicator $\boldsymbol{\xi}^* \in \Theta^*$, we prove that $\boldsymbol{\mu}^*$ is identifiable. By the definition of $\boldsymbol{\mu}^*$, that is,

$$\boldsymbol{\mu}^* = \arg\max_{\boldsymbol{\mu} \in \mathcal{H}_k^{\boldsymbol{\xi}^*}} L(\boldsymbol{\mu}, \boldsymbol{\xi}^*),$$

then by taking $r > 0$ such that $\boldsymbol{\mu}^* \in \mathcal{B}(\boldsymbol{\mu}^*, r)$, we have

$$\boldsymbol{\mu}^* = \arg\max_{\boldsymbol{\mu} \in \mathcal{B}(\boldsymbol{\mu}^*, r)} L(\boldsymbol{\mu}, \boldsymbol{\xi}^*),$$

where $\mathcal{B}(\boldsymbol{\mu}^*, r) \subset \mathcal{H}_k^{\boldsymbol{\xi}^*}$ is a closed ball centered at $\boldsymbol{\mu}^*$ and with radius r. On the other hand, Lemma 4.4 of Levrard (2015) ensures that there exists $M > 0$

and $\boldsymbol{\nu}^* \in \mathcal{B}(0, M+r)\backslash\mathcal{B}^o(\boldsymbol{\mu}^*, r)$, such that

$$L(\boldsymbol{\nu}^*, \boldsymbol{\xi}^*) = \sup_{\boldsymbol{\mu}\in\mathcal{H}_k^{\boldsymbol{\xi}^*}\backslash\mathcal{B}^o(\boldsymbol{\mu}^*,r)} L(\boldsymbol{\mu}, \boldsymbol{\xi}^*),$$

where $\mathcal{B}(0, M+r) \subset \mathcal{H}_k^{\boldsymbol{\xi}^*}$ is centered at 0 and with radius $M+r$, and $\mathcal{B}^o(\boldsymbol{\mu}^*, r)$ is the interior of $\mathcal{B}(\boldsymbol{\mu}^*, r)$. In other words, away from $\boldsymbol{\mu}^*$, there is no sequence of $\boldsymbol{\mu}$ such that $L(\boldsymbol{\mu}, \boldsymbol{\xi}^*)$ tends to $L(\boldsymbol{\mu}^*, \boldsymbol{\xi}^*)$. Therefore, we have for any $\epsilon > 0$

$$L(\boldsymbol{\mu}^*, \boldsymbol{\xi}^*) > \sup\{L(\boldsymbol{\mu}, \boldsymbol{\xi}^*) \mid D(\boldsymbol{\mu}, \boldsymbol{\mu}^*) > \epsilon;\ \mu_l \in \mathcal{H}^{\boldsymbol{\xi}^*}, \forall l = 1, \ldots, k\}. \quad (2.21)$$

Secondly, for a fixed $\boldsymbol{\zeta}$ belonging to Θ^* but not an optimal indicator, similar to the proof of Lemma 2.2, we can obtain that $\boldsymbol{\mu}^*$ uniquely maximizes $L(\boldsymbol{\mu}, \boldsymbol{\zeta})$ about $\boldsymbol{\mu}$ and $L(\boldsymbol{\mu}^*, \boldsymbol{\zeta}) < L(\boldsymbol{\mu}^*, \boldsymbol{\xi}^*)$. Also, similar to the above analysis, we can obtain the identifiablity of $\boldsymbol{\mu}^*$ for such $\boldsymbol{\zeta}$.

Finally, combining Condition 2.4 leads to the identifiability of $(\boldsymbol{\mu}^*, \boldsymbol{\xi}^*)$ in the following sense: For any $\epsilon > 0$,

$$L(\boldsymbol{\mu}^*, \boldsymbol{\xi}^*) > \sup\left\{L(\boldsymbol{\mu}, \boldsymbol{\xi}) \mid D(\boldsymbol{\mu}, \boldsymbol{\mu}^*) > \epsilon \text{ or } \boldsymbol{\xi} \notin \Theta^*;\ \mu_l \in \mathcal{H}^{\boldsymbol{\xi}}, \forall l\right\}.$$

That is, for any $\epsilon > 0$, there exists $\tilde{\epsilon} > 0$ depending on ϵ, such that $L(\boldsymbol{\mu}^*, \boldsymbol{\xi}^*) > L(\boldsymbol{\mu}, \boldsymbol{\xi}) + \tilde{\epsilon}$ holds for any $(\boldsymbol{\mu}, \boldsymbol{\xi})$ satisfying $D(\boldsymbol{\mu}, \boldsymbol{\mu}^*) > \epsilon$ or $\boldsymbol{\xi} \notin \Theta^*$. It implies that the event $\{D(\hat{\boldsymbol{\mu}}, \boldsymbol{\mu}^*) > \epsilon \text{ or } \hat{\boldsymbol{\xi}} \notin \Theta^*\}$ is included in the event $\{L(\boldsymbol{\mu}^*, \boldsymbol{\xi}^*) > L(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\xi}}) + \tilde{\epsilon}\}$, and thus

$$\Pr\left(D(\hat{\boldsymbol{\mu}}, \boldsymbol{\mu}^*) > \epsilon \text{ or } \hat{\boldsymbol{\xi}} \notin \Theta^*\right) \leq \Pr\left(L(\boldsymbol{\mu}^*, \boldsymbol{\xi}^*) > L(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\xi}}) + \tilde{\epsilon}\right).$$

Based on (i), for this given $\tilde{\epsilon}$, we have $\lim_{n\to\infty} \Pr\left(L(\boldsymbol{\mu}^*, \boldsymbol{\xi}^*) - L(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\xi}}) > \tilde{\epsilon}\right) = 0$. It follows that $\lim_{n\to\infty} \Pr\left(D(\hat{\boldsymbol{\mu}}, \boldsymbol{\mu}^*) > \epsilon \text{ or } \hat{\boldsymbol{\xi}} \notin \Theta^*\right) = 0$.

2.7.5 Proof of Theorem 2.2

In this section, we prove Theorem 2.2 about the convergence in probability of $(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\xi}})$ to $(\boldsymbol{\mu}^{**}, \boldsymbol{\xi}^{**})$ under assumptions on λ_n.

For any $n \in \mathbb{N}_+$, $(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\xi}})$ maximizes $\widehat{L}_n(\boldsymbol{\mu}, \boldsymbol{\xi})$, which means $\widehat{L}_n(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\xi}}) \geq \widehat{L}_n(\tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\xi}})$. If $\|\hat{\boldsymbol{\xi}}\|_0 > d_0$, then we have

$$\frac{\widetilde{L}_n^{(\mathrm{SKKM})}(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\xi}}) - \widetilde{L}_n^{(\mathrm{SKKM})}(\tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\xi}})}{\|\hat{\boldsymbol{\xi}}\|_0 - \|\tilde{\boldsymbol{\xi}}\|_0} \geq \lambda_n.$$

Moreover, since $\|\tilde{\boldsymbol{\xi}}\|_0 = d_0$, then by the definition of $\nabla_n^+(d_0)$ (Eq.(2.16)), we have

$$\frac{\widetilde{L}_n^{(\mathrm{SKKM})}(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\xi}}) - \widetilde{L}_n^{(\mathrm{SKKM})}(\tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\xi}})}{\|\hat{\boldsymbol{\xi}}\|_0 - \|\tilde{\boldsymbol{\xi}}\|_0} \leq \nabla_n^+(d_0).$$

It implies that $\Pr\left(\|\hat{\boldsymbol{\xi}}\|_0 > d_0\right) \leq \Pr\left(\lambda_n \leq \nabla_n^+(d_0)\right)$. Therefore, by the assumption $\lim_{n\to\infty} \Pr(\lambda_n > \nabla_n^+(d_0)) = 1$, we obtain

$$\lim_{n\to\infty} \Pr\left(\|\hat{\boldsymbol{\xi}}\|_0 > d_0\right) = 0. \tag{2.22}$$

According to (ii) of Theorem 2.1, we know for any $\epsilon > 0$, $\delta > 0$, there exists $N_1 \in \mathbb{N}$ such that for $n \geq N_1$, it holds that

$$\Pr\left(D(\hat{\boldsymbol{\mu}}, \boldsymbol{\mu}^*) \leq \epsilon \text{ and } \hat{\boldsymbol{\xi}} \in \Theta^*\right) \geq 1 - \frac{\delta}{2}. \tag{2.23}$$

According to Eq.(2.22), there exists $N_2 \in \mathbb{N}$ such that for $n \geq N_2$, it holds that $\Pr\left(\|\hat{\boldsymbol{\xi}}\|_0 > d_0\right) < \delta/2$, which follows that

$$\Pr\left(D(\hat{\boldsymbol{\mu}}, \boldsymbol{\mu}^*) \leq \epsilon \text{ and } \hat{\boldsymbol{\xi}} \in \Theta^* \text{ and } \|\hat{\boldsymbol{\xi}}\|_0 > d_0\right) < \frac{\delta}{2}. \tag{2.24}$$

Because

$$\begin{aligned}
&\Pr\left(D(\hat{\boldsymbol{\mu}}, \boldsymbol{\mu}^*) \leq \epsilon \text{ and } \hat{\boldsymbol{\xi}} \in \Theta^*\right) \\
&= \Pr\left(D(\hat{\boldsymbol{\mu}}, \boldsymbol{\mu}^*) \leq \epsilon \text{ and } \hat{\boldsymbol{\xi}} \in \Theta^* \text{ and } \|\hat{\boldsymbol{\xi}}\|_0 > d_0\right) \\
&\quad + \Pr\left(D(\hat{\boldsymbol{\mu}}, \boldsymbol{\mu}^*) \leq \epsilon \text{ and } \hat{\boldsymbol{\xi}} \in \Theta^* \text{ and } \|\hat{\boldsymbol{\xi}}\|_0 = d_0\right),
\end{aligned}$$

then by combining Eq.(2.23) and Eq.(2.24), for any $n > \max\{N_1, N_2\}$, we have

$$\Pr\left(D(\hat{\boldsymbol{\mu}}, \boldsymbol{\mu}^*) \leq \epsilon \text{ and } \hat{\boldsymbol{\xi}} \in \Theta^* \text{ and } \|\hat{\boldsymbol{\xi}}\|_0 = d_0\right) > 1 - \delta,$$

which completes the proof.

2.7.6 Some Lemmas and proofs

Lemma 2.4. *For any $\tilde{\epsilon} > 0$, there exists $N \in \mathbb{N}_+$ such that for any $n \geq N$,*

$$\Pr\left(\sup_{\substack{\boldsymbol{\xi}\in\{0,1\}^p \\ \mu_l\in\mathcal{H}^{\boldsymbol{\xi}},\forall l=1,\ldots,k}} \left|\widehat{L}_n(\boldsymbol{\mu},\boldsymbol{\xi}) - L(\boldsymbol{\mu},\boldsymbol{\xi})\right| > \tilde{\epsilon}\right) \leq 4\exp\left(-\left[\frac{(\tilde{\epsilon}-p\lambda_n)\sqrt{n}-C_1}{C_2}\right]^2\right),$$

where

$$C_1 = 4k(2^{p+1}+1)c_U \text{ and } C_2 = (6\sqrt{2}+36c_U)c_U. \tag{2.25}$$

Proof. We first define two function classes on $\mathcal{X}$ be

$$\mathcal{G}_k = \left\{ \|\psi^{\boldsymbol{\xi}}(\cdot) - \mu_0^{\boldsymbol{\xi}}\|_{\mathcal{H}}^2 - \min_{l=1,\ldots,k}\|\psi^{\boldsymbol{\xi}}(\cdot) - \mu_l\|_{\mathcal{H}}^2 \,\middle|\, \boldsymbol{\xi}\in\{0,1\}^p, \mu_l\in\mathcal{H}^{\boldsymbol{\xi}}, \|\mu_l\|_{\mathcal{H}}^2 \leq c_U, \forall l=1,\ldots,k \right\}$$

$$\mathcal{G} = \left\{ \|\psi^{\boldsymbol{\xi}}(\cdot) - \mu_0^{\boldsymbol{\xi}}\|_{\mathcal{H}}^2 - \|\psi^{\boldsymbol{\xi}}(\cdot) - \mu\|_{\mathcal{H}}^2 \,\middle|\, \boldsymbol{\xi}\in\{0,1\}^p, \mu\in\mathcal{H}^{\boldsymbol{\xi}}, \|\mu\|_{\mathcal{H}}^2 \leq c_U \right\},$$

where $\mu_0^{\boldsymbol{\xi}} = \mathbb{E}_{X\sim\mathbb{P}}[\psi^{\boldsymbol{\xi}}(X)]$. Recall $\psi^{\boldsymbol{\xi}}(x) = h(\cdot\circ\boldsymbol{\xi}, x\circ\boldsymbol{\xi})$, then we have $\|\psi^{\boldsymbol{\xi}}(x)\|_{\mathcal{H}}^2 \leq c_U$ for any $x\in\mathcal{X}$. For the sample $\{X_1,\ldots,X_n\}$, denote the empirical Rademacher complexity by $\widehat{\mathfrak{R}}$. Then, for any function $g_k\in\mathcal{G}_k : \mathcal{X}\to\mathbb{R}$, according to Theorem 3.3 in Mohri et al. (2018) and Theorem 12 of Bartlett & Mendelson (2002) that $\hat{\mathfrak{R}}(\mathcal{G}_k) \leq k\cdot\hat{\mathfrak{R}}(\mathcal{G})$, the following holds for any $\delta_1\in(0,1)$:

$$\Pr\left(\mathbb{E}_{X\sim\mathbb{P}}[g_k(X)] - \frac{1}{n}\sum_{i=1}^n g_k(X_i) \leq 2k\hat{\mathfrak{R}}(\mathcal{G}) + 6c_U\sqrt{\frac{\log(2/\delta_1)}{2n}}\right) > 1-\delta_1. \tag{2.26}$$

We next derive the upper bound of $\hat{\mathfrak{R}}(\mathcal{G})$. Let $\boldsymbol{\epsilon} = \{\epsilon_i\}_{i=1}^n$ be independent random variables that take the value ±1 with equal probability $\frac{1}{2}$, then the empirical Rademacher complexity of $\mathcal{G}$ is given by

$$\begin{aligned}
\hat{\mathfrak{R}}(\mathcal{G}) &= \mathbb{E}_{\boldsymbol{\epsilon}}\left[\sup_{g\in\mathcal{G}} \frac{1}{n}\sum_{i=1}^{n} \epsilon_i g(X_i)\right] \\
&= \mathbb{E}_{\boldsymbol{\epsilon}}\left[\sup_{\substack{\boldsymbol{\xi}\in\{0,1\}^p \\ \|\mu\|_{\mathcal{H}}^2\le c_U}} \frac{1}{n}\sum_{i=1}^{n} \epsilon_i \left(\|\psi^{\boldsymbol{\xi}}(X_i)-\mu_0^{\boldsymbol{\xi}}\|_{\mathcal{H}}^2 - \|\psi^{\boldsymbol{\xi}}(X_i)-\mu\|_{\mathcal{H}}^2\right)\right] \\
&= \mathbb{E}_{\boldsymbol{\epsilon}}\left[\sup_{\substack{\boldsymbol{\xi}\in\{0,1\}^p \\ \|\mu\|_{\mathcal{H}}^2\le c_U}} \frac{1}{n}\sum_{i=1}^{n} \epsilon_i \left(\|\mu_0^{\boldsymbol{\xi}}\|_{\mathcal{H}}^2 - \|\mu\|_{\mathcal{H}}^2 - 2\langle\psi^{\boldsymbol{\xi}}(X_i),\mu_0^{\boldsymbol{\xi}}\rangle_{\mathcal{H}} + 2\langle\psi^{\boldsymbol{\xi}}(X_i),\mu\rangle_{\mathcal{H}}\right)\right] \\
&\le \mathbb{E}_{\boldsymbol{\epsilon}}\left[\sup_{\substack{\boldsymbol{\xi}\in\{0,1\}^p \\ \|\mu\|_{\mathcal{H}}^2\le c_U}} \frac{1}{n}\sum_{i=1}^{n} \epsilon_i \|\mu_0^{\boldsymbol{\xi}}\|_{\mathcal{H}}^2\right] + \mathbb{E}_{\boldsymbol{\epsilon}}\left[\sup_{\substack{\boldsymbol{\xi}\in\{0,1\}^p \\ \|\mu\|_{\mathcal{H}}^2\le c_U}} \frac{1}{n}\sum_{i=1}^{n} \epsilon_i \|\mu\|_{\mathcal{H}}^2\right] \\
&\quad + 2\cdot\mathbb{E}_{\boldsymbol{\epsilon}}\left[\sup_{\substack{\boldsymbol{\xi}\in\{0,1\}^p \\ \|\mu\|_{\mathcal{H}}^2\le c_U}} \left|\frac{1}{n}\sum_{i=1}^{n} \epsilon_i \langle\psi^{\boldsymbol{\xi}}(X_i),\mu_0^{\boldsymbol{\xi}}\rangle_{\mathcal{H}}\right|\right] \\
&\quad + 2\cdot\mathbb{E}_{\boldsymbol{\epsilon}}\left[\sup_{\substack{\boldsymbol{\xi}\in\{0,1\}^p \\ \|\mu\|_{\mathcal{H}}^2\le c_U}} \left|\frac{1}{n}\sum_{i=1}^{n} \epsilon_i \langle\psi^{\boldsymbol{\xi}}(X_i),\mu\rangle_{\mathcal{H}}\right|\right].
\end{aligned}$$

For the second term, we have

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{\epsilon}}\left[\sup_{\substack{\boldsymbol{\xi}\in\{0,1\}^p \\ \|\mu\|_{\mathcal{H}}^2\le c_U}} \frac{1}{n}\sum_{i=1}^{n} \epsilon_i \|\mu\|_{\mathcal{H}}^2\right] \\
&\le \frac{1}{n}\mathbb{E}_{\boldsymbol{\epsilon}}\left[\left|\sum_{i=1}^{n}\epsilon_i\right| \cdot \sup_{\|\mu\|_{\mathcal{H}}^2\le c_U} \|\mu\|_{\mathcal{H}}^2\right] \le \frac{c_U}{n}\mathbb{E}_{\boldsymbol{\epsilon}}\left[\left|\sum_{i=1}^{n}\epsilon_i\right|\right] \\
&\le \frac{c_U}{n}\cdot\left\{\sum_{i,i'=1}^{n}\mathbb{E}_{\boldsymbol{\epsilon}}[\epsilon_i\epsilon_{i'}]\right\}^{1/2} = \frac{c_U}{n}\cdot\sqrt{n} = \frac{c_U}{\sqrt{n}}.
\end{aligned}$$

For the fourth term, we have

$$
\begin{aligned}
&2 \cdot \mathbb{E}_{\boldsymbol{\epsilon}}\left[\sup_{\substack{\boldsymbol{\xi}\in\{0,1\}^p \\ \|\mu\|_{\mathcal{H}}^2\le c_U}} \left|\frac{1}{n}\sum_{i=1}^n \epsilon_i \langle \psi^{\boldsymbol{\xi}}(X_i), \mu\rangle_{\mathcal{H}}\right|\right] = \frac{2}{n}\mathbb{E}_{\boldsymbol{\epsilon}}\left[\sup_{\substack{\boldsymbol{\xi}\in\{0,1\}^p \\ \|\mu\|_{\mathcal{H}}^2\le c_U}} \left|\left\langle \sum_{i=1}^n \epsilon_i \psi^{\boldsymbol{\xi}}(X_i), \mu\right\rangle_{\mathcal{H}}\right|\right] \\
&\le \frac{2}{n}\mathbb{E}_{\boldsymbol{\epsilon}}\left[\sup_{\substack{\boldsymbol{\xi}\in\{0,1\}^p \\ \|\mu\|_{\mathcal{H}}^2\le c_U}} \|\mu\|_{\mathcal{H}} \cdot \left\|\sum_{i=1}^n \epsilon_i \psi^{\boldsymbol{\xi}}(X_i)\right\|_{\mathcal{H}}\right] \le \frac{2\sqrt{c_U}}{n}\mathbb{E}_{\boldsymbol{\epsilon}}\left[\sup_{\boldsymbol{\xi}\in\{0,1\}^p} \left\|\sum_{i=1}^n \epsilon_i \psi^{\boldsymbol{\xi}}(X_i)\right\|_{\mathcal{H}}\right] \\
&\le \frac{2\sqrt{c_U}}{n}\mathbb{E}_{\boldsymbol{\epsilon}}\left[\sum_{\boldsymbol{\xi}\in\{0,1\}^p} \left\|\sum_{i=1}^n \epsilon_i \psi^{\boldsymbol{\xi}}(X_i)\right\|_{\mathcal{H}}\right] = \frac{2\sqrt{c_U}}{n}\sum_{\boldsymbol{\xi}\in\{0,1\}^p} \mathbb{E}_{\boldsymbol{\epsilon}}\left[\left\|\sum_{i=1}^n \epsilon_i \psi^{\boldsymbol{\xi}}(X_i)\right\|_{\mathcal{H}}\right] \\
&\le \frac{2\sqrt{c_U}}{n}\sum_{\boldsymbol{\xi}\in\{0,1\}^p} \left\{\mathbb{E}_{\boldsymbol{\epsilon}}\left[\left\|\sum_{i=1}^n \epsilon_i \psi^{\boldsymbol{\xi}}(X_i)\right\|_{\mathcal{H}}^2\right]\right\}^{1/2} \\
&= \frac{2\sqrt{c_U}}{n}\sum_{\boldsymbol{\xi}\in\{0,1\}^p} \left\{\mathbb{E}_{\boldsymbol{\epsilon}}\left[\sum_{i,i'=1}^n \epsilon_i\epsilon_{i'} h^{\boldsymbol{\xi}}(X_i, X_{i'})\right]\right\}^{1/2} \\
&\le \frac{2\sqrt{c_U}}{n}\sum_{\boldsymbol{\xi}\in\{0,1\}^p} \left\{c_U \sum_{i,i'=1}^n \mathbb{E}_{\boldsymbol{\epsilon}}\left[\epsilon_i\epsilon_{i'}\right]\right\}^{1/2} = \frac{2^{p+1}c_U}{n}\left\{\sum_{i,i'=1}^n \mathbb{E}_{\boldsymbol{\epsilon}}\left[\epsilon_i\epsilon_{i'}\right]\right\}^{1/2} \\
&\le \frac{2^{p+1}c_U}{n} \cdot \sqrt{n} = \frac{2^{p+1}c_U}{\sqrt{n}}.
\end{aligned}
$$

Because for any $\boldsymbol{\xi} \in \{0,1\}^p$, we have $\mu_0^{\boldsymbol{\xi}} \in \mathcal{H}$ and

$$
\|\mu_0^{\boldsymbol{\xi}}\|_{\mathcal{H}}^2 = \left\langle \int_{\mathcal{X}} \psi^{\boldsymbol{\xi}}(x)\, d\mathbb{P}(x), \int_{\mathcal{X}} \psi^{\boldsymbol{\xi}}(\tilde{x})\, d\mathbb{P}(\tilde{x})\right\rangle_{\mathcal{H}} = \mathbb{E}_X\mathbb{E}_{\tilde{X}}[h(X\circ\boldsymbol{\xi}, \tilde{X}\circ\boldsymbol{\xi})] \le c_U,
$$

then for the first term and third term, we can similarly get

$$
\begin{aligned}
\mathbb{E}_{\boldsymbol{\epsilon}}\left[\sup_{\substack{\boldsymbol{\xi}\in\{0,1\}^p \\ \|\mu\|_{\mathcal{H}}^2\le c_U}} \frac{1}{n}\sum_{i=1}^n \epsilon_i \|\mu_0^{\boldsymbol{\xi}}\|_{\mathcal{H}}^2\right] &\le \frac{1}{n}\mathbb{E}_{\boldsymbol{\epsilon}}\left[\left|\sum_{i=1}^n \epsilon_i\right| \cdot \sup_{\boldsymbol{\xi}\in\{0,1\}^p} \|\mu_0^{\boldsymbol{\xi}}\|_{\mathcal{H}}^2\right] \\
&\le \frac{c_U}{n} \cdot \sqrt{n} = \frac{c_U}{\sqrt{n}},
\end{aligned}
$$

and

$$\begin{aligned}
2 \cdot \mathbb{E}_{\boldsymbol{\epsilon}} \left[\sup_{\substack{\boldsymbol{\xi} \in \{0,1\}^p \\ \|\mu\|_{\mathcal{H}}^2 \le c_U}} \left| \frac{1}{n} \sum_{i=1}^n \epsilon_i \langle \psi^{\boldsymbol{\xi}}(X_i), \mu_0^{\boldsymbol{\xi}} \rangle_{\mathcal{H}} \right| \right] &\le \frac{2}{n} \mathbb{E}_{\boldsymbol{\epsilon}} \left[\sup_{\boldsymbol{\xi} \in \{0,1\}^p} \|\mu_0^{\boldsymbol{\xi}}\|_{\mathcal{H}} \cdot \left\| \sum_{i=1}^n \epsilon_i \psi^{\boldsymbol{\xi}}(X_i) \right\|_{\mathcal{H}} \right] \\
&\le \frac{2\sqrt{c_U}}{n} \mathbb{E}_{\boldsymbol{\epsilon}} \left[\sup_{\boldsymbol{\xi} \in \{0,1\}^p} \left\| \sum_{i=1}^n \epsilon_i \psi^{\boldsymbol{\xi}}(X_i) \right\|_{\mathcal{H}} \right] \\
&\le \frac{2^{p+1} c_U}{\sqrt{n}}.
\end{aligned}$$

Therefore, we have

$$\hat{\mathfrak{R}}(\mathcal{G}) \le \frac{2(2^{p+1}+1)c_U}{\sqrt{n}}. \tag{2.27}$$

Combining Eq. (2.26) and Eq. (2.27), and using the symmetry, we have for any $\delta_1 \in (0,1)$,

$$\Pr\left(\left| \mathbb{E}_{X \sim \mathbb{P}}[g_k(X)] - \frac{1}{n} \sum_{i=1}^n g_k(X_i) \right| \le \frac{4k(2^{p+1}+1)c_U}{\sqrt{n}} + 12 c_U \sqrt{\frac{\log(2/\delta_1)}{2n}} \right) \\ > 1 - \delta_1. \tag{2.28}$$

Secondly, we derive the bound for $\|\mu_0^{\boldsymbol{\xi}} - \hat{\mu}_0^{\boldsymbol{\xi}}\|_{\mathcal{H}}^2$. Recall that for a given $\boldsymbol{\xi}$, the $\boldsymbol{\mu}_0^{\boldsymbol{\xi}} = \mathbb{E}_X[\psi^{\boldsymbol{\xi}}(X)]$ is the population mean and the $\hat{\boldsymbol{\mu}}_0^{\boldsymbol{\xi}} = \frac{1}{n}\sum_{i=1}^n \psi^{\boldsymbol{\xi}}(X_i)$ is the sample average. Since we have

$$\|\mu_0^{\boldsymbol{\xi}} - \hat{\mu}_0^{\boldsymbol{\xi}}\|_{\mathcal{H}} = \left\| \frac{1}{n} \sum_{i=1}^n \left\{ \psi^{\boldsymbol{\xi}}(X_i) - \mathbb{E}_X[\psi^{\boldsymbol{\xi}}(X)] \right\} \right\|_{\mathcal{H}},$$

and for any $i = 1, \ldots, n$,

$$\|\psi^{\boldsymbol{\xi}}(X_i)\|_\infty = \sup_{x \in \mathcal{X}} \|h^{\boldsymbol{\xi}}(x, X_i)\|_{\mathcal{H}} \le c_U,$$

then according to Corollary 6.15 of Steinwart & Christmann (2008), for any $\tau > 0$, it holds that

$$\Pr\left(\|\mu_0^{\boldsymbol{\xi}} - \hat{\mu}_0^{\boldsymbol{\xi}}\|_{\mathcal{H}} \le c_U \sqrt{\frac{2\tau}{n}} + c_U \frac{1}{n} + \frac{4 c_U \tau}{3n} \right) > 1 - \exp(-\tau).$$

Since for $1/n \leq \tau \leq n$, it holds that

$$\begin{aligned}
& c_U\sqrt{\frac{2\tau}{n}} + c_U\frac{1}{n} + \frac{4c_U\tau}{3n} \\
&= c_U\sqrt{\frac{\tau}{n}}\left(\sqrt{2} + \frac{1}{\sqrt{\tau n}} + \frac{4}{3}\sqrt{\frac{\tau}{n}}\right) \\
&\leq c_U\sqrt{\frac{\tau}{n}}\left(\sqrt{2} + 1 + \frac{4}{3}\right) \\
&\leq 6c_U\sqrt{\frac{\tau}{n}}
\end{aligned}$$

then have

$$\Pr\left(\|\mu_0^{\boldsymbol{\xi}} - \hat{\mu}_0^{\boldsymbol{\xi}}\|_{\mathcal{H}} \leq 6c_U\sqrt{\frac{\tau}{n}}\right) > 1 - \exp(-\tau).$$

It is equivalent to: For any $\delta_2 \in (e^{-n}, e^{-1/n})$, it holds that

$$\Pr\left(\|\mu_0^{\boldsymbol{\xi}} - \hat{\mu}_0^{\boldsymbol{\xi}}\|_{\mathcal{H}} \leq 6c_U\sqrt{\frac{\log(1/\delta_2)}{n}}\right) > 1 - \delta_2.$$

It also means that

$$\Pr\left(\|\mu_0^{\boldsymbol{\xi}} - \hat{\mu}_0^{\boldsymbol{\xi}}\|_{\mathcal{H}}^2 \leq 36c_U^2\frac{\log(1/\delta_2)}{n}\right) > 1 - \delta_2. \tag{2.29}$$

Finally, we derive the uniform bound for $\left|\widehat{L}_n(\boldsymbol{\mu}, \boldsymbol{\xi}) - L(\boldsymbol{\mu}, \boldsymbol{\xi})\right|$. Since for any $(\boldsymbol{\mu}, \boldsymbol{\xi})$,

$$\begin{aligned}
\left|\widehat{L}_n(\boldsymbol{\mu}, \boldsymbol{\xi}) - L(\boldsymbol{\mu}, \boldsymbol{\xi})\right| &= \left|\frac{1}{n}\sum_{i=1}^{n} g_k(X_i) - \mathbb{E}_X[g_k(X)] - \|\mu_0^{\boldsymbol{\xi}} - \hat{\mu}_0^{\boldsymbol{\xi}}\|_{\mathcal{H}}^2 - \lambda_n\|\boldsymbol{\xi}\|_0\right| \\
&\leq \left|\frac{1}{n}\sum_{i=1}^{n} g_k(X_i) - \mathbb{E}_X[g_k(X)]\right| + \|\mu_0^{\boldsymbol{\xi}} - \hat{\mu}_0^{\boldsymbol{\xi}}\|_{\mathcal{H}}^2 + p\lambda_n,
\end{aligned}$$

then for any $\delta \in (2e^{-n}, 2e^{-1/n})$, by combining Eq. (2.28) and Eq. (2.29) with $\delta_1 = \delta_2 = \delta/2$, we have

$$\Pr\left(\left|\widehat{L}_n(\boldsymbol{\mu}, \boldsymbol{\xi}) - L(\boldsymbol{\mu}, \boldsymbol{\xi})\right| \leq \frac{4k(2^{p+1}+1)c_U}{\sqrt{n}} + 12c_U\sqrt{\frac{\log(4/\delta)}{2n}} + 36c_U^2\frac{\log(2/\delta)}{n} + p\lambda_n\right)$$
$$> 1 - \delta.$$

Because $\log(2/\delta) < n$ for $\delta \in (2e^{-n}, 2e^{-1/n})$, we have $\log(2/\delta)/n \leq \sqrt{\log(2/\delta)/n} \leq \sqrt{\log(4/\delta)/n}$. It follows that

$$\Pr\left(\left|\widehat{L}_n(\boldsymbol{\mu},\boldsymbol{\xi}) - L(\boldsymbol{\mu},\boldsymbol{\xi})\right| \leq \frac{C_1 + C_2\sqrt{\log(4/\delta)}}{\sqrt{n}} + p\lambda_n\right) > 1-\delta,$$

where

$$C_1 = 4k(2^{p+1}+1)c_U \text{ and } C_2 = (6\sqrt{2}+36c_U)c_U.$$

Consequently, for any $\tilde{\epsilon} > 0$, there exists $N \in \mathbb{N}_+$ such that for any $n \geq N$, it holds that

$$2e^{-n} < 4\exp\left(-\left[\frac{(\tilde{\epsilon}-p\lambda_n)\sqrt{n}-C_1}{C_2}\right]^2\right) < 2e^{-\frac{1}{n}},$$

and then we have

$$\Pr\left(\left|\widehat{L}_n(\boldsymbol{\mu},\boldsymbol{\xi}) - L(\boldsymbol{\mu},\boldsymbol{\xi})\right| \leq \tilde{\epsilon}\right) > 1 - 4\exp\left(-\left[\frac{(\tilde{\epsilon}-p\lambda_n)\sqrt{n}-C_1}{C_2}\right]^2\right),$$

which completes the proof. □

Lemma 2.5. *For any $\delta \in (2e^{-n}, 2e^{-1/n})$, we have*

$$\Pr\left(\sup_{\substack{\boldsymbol{\xi}\in\{0,1\}^p\\ \boldsymbol{\mu}\in\mathcal{H}^{\boldsymbol{\xi}}}}\left|\widetilde{L}_n^{(\mathrm{SKKM})}(\boldsymbol{\mu},\boldsymbol{\xi}) - L(\boldsymbol{\mu},\boldsymbol{\xi})\right| \leq \frac{C_1 + C_2\sqrt{\log(4/\delta)}}{\sqrt{n}}\right) > 1-\delta,$$

where

$$C_1 = 4k(2^{p+1}+1)c_U \text{ and } C_2 = (6\sqrt{2}+36c_U)c_U.$$

Proof. Since $\widehat{L}_n(\boldsymbol{\mu},\boldsymbol{\xi}) = \widetilde{L}_n^{(\mathrm{SKKM})}(\boldsymbol{\mu},\boldsymbol{\xi}) + \lambda_n\|\boldsymbol{\xi}\|_0$, by taking $\lambda_n = 0$ and using the same proof of Lemma 2.4, we immediately complete the proof. □

Lemma 2.6. *Under Conditions 2.1-2.4, for $\nabla_n^+(d_0)$ defined in Eq.(2.16), we have $\nabla_n^+(d_0) = O_P(1/\sqrt{n})$.*

Proof. Recall that

$$(\tilde{\boldsymbol{\mu}},\tilde{\boldsymbol{\xi}}) = \underset{\substack{\boldsymbol{\xi}\in\{0,1\}^p,\|\boldsymbol{\xi}\|_0\leq d_0\\ \boldsymbol{\mu}\in\mathcal{H}_k^{\boldsymbol{\xi}}}}{\arg\max}\ \widetilde{L}_n^{(\mathrm{SKKM})}(\boldsymbol{\mu},\boldsymbol{\xi}).$$

At first, we note that

$$\nabla_n^+(d_0) \le \max_{\|\boldsymbol{\xi}\|_0 > d_0} \left\{ \widetilde{L}_n^{(\mathrm{SKKM})}(\tilde{\boldsymbol{\mu}}^{\boldsymbol{\xi}}, \boldsymbol{\xi}) - \widetilde{L}_n^{(\mathrm{SKKM})}(\tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\xi}}) \right\},$$

then, it suffices to prove: For any $\boldsymbol{\xi} \in \{0,1\}^p$ with $\|\boldsymbol{\xi}\|_0 > d_0$, the following holds

$$\widetilde{L}_n^{(\mathrm{SKKM})}(\tilde{\boldsymbol{\mu}}^{\boldsymbol{\xi}}, \boldsymbol{\xi}) - \widetilde{L}_n^{(\mathrm{SKKM})}(\tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\xi}}) = O_P(1/\sqrt{n}). \tag{2.30}$$

Next, we consider some fixed $\boldsymbol{\xi} \in \{0,1\}^p$ with $\|\boldsymbol{\xi}\|_0 = t$, where $t \in \{d_0+1, \cdots, p\}$. Since

$$\begin{aligned} &\widetilde{L}_n^{(\mathrm{SKKM})}(\tilde{\boldsymbol{\mu}}^{\boldsymbol{\xi}}, \boldsymbol{\xi}) - \widetilde{L}_n^{(\mathrm{SKKM})}(\tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\xi}}) \\ &= \underbrace{\widetilde{L}_n^{(\mathrm{SKKM})}(\tilde{\boldsymbol{\mu}}^{\boldsymbol{\xi}}, \boldsymbol{\xi}) - L(\boldsymbol{\mu}^{**}, \boldsymbol{\xi}^{**})}_{\text{(I)}} + \underbrace{L(\boldsymbol{\mu}^{**}, \boldsymbol{\xi}^{**}) - \widetilde{L}_n^{(\mathrm{SKKM})}(\tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\xi}})}_{\text{(II)}}, \end{aligned}$$

we will bound the two parts respectively. Because

$$\begin{aligned} \text{(I)} &= \widetilde{L}_n^{(\mathrm{SKKM})}(\tilde{\boldsymbol{\mu}}^{\boldsymbol{\xi}}, \boldsymbol{\xi}) - L(\boldsymbol{\mu}^{**}, \boldsymbol{\xi}^{**}) \\ &= \left[\widetilde{L}_n^{(\mathrm{SKKM})}(\tilde{\boldsymbol{\mu}}^{\boldsymbol{\xi}}, \boldsymbol{\xi}) - L(\tilde{\boldsymbol{\mu}}^{\boldsymbol{\xi}}, \boldsymbol{\xi})\right] + \left[L(\tilde{\boldsymbol{\mu}}^{\boldsymbol{\xi}}, \boldsymbol{\xi}) - L(\boldsymbol{\mu}^{**}, \boldsymbol{\xi}^{**})\right] \\ &\le \sup_{\substack{\boldsymbol{\eta}\in\{0,1\}^p \\ \boldsymbol{\nu}\in\mathcal{H}_k^{\boldsymbol{\eta}}}} \left|\widetilde{L}_n^{(\mathrm{SKKM})}(\boldsymbol{\nu}, \boldsymbol{\eta}) - L(\boldsymbol{\nu}, \boldsymbol{\eta})\right| + 0, \end{aligned}$$

and

$$\begin{aligned} \text{(II)} &= L(\boldsymbol{\mu}^{**}, \boldsymbol{\xi}^{**}) - \widetilde{L}_n^{(\mathrm{SKKM})}(\tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\xi}}) \\ &= \left[L(\boldsymbol{\mu}^{**}, \boldsymbol{\xi}^{**}) - \widetilde{L}_n^{(\mathrm{SKKM})}(\boldsymbol{\mu}^{**}, \boldsymbol{\xi}^{**})\right] + \left[\widetilde{L}_n^{(\mathrm{SKKM})}(\boldsymbol{\mu}^{**}, \boldsymbol{\xi}^{**}) - \widetilde{L}_n^{(\mathrm{SKKM})}(\tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\xi}})\right] \\ &\le \sup_{\substack{\boldsymbol{\eta}\in\{0,1\}^p \\ \boldsymbol{\nu}\in\mathcal{H}_k^{\boldsymbol{\eta}}}} \left|\widetilde{L}_n^{(\mathrm{SKKM})}(\boldsymbol{\nu}, \boldsymbol{\eta}) - L(\boldsymbol{\nu}, \boldsymbol{\eta})\right| + 0, \end{aligned}$$

then we have

$$\widetilde{L}_n^{(\mathrm{SKKM})}(\tilde{\boldsymbol{\mu}}^{\boldsymbol{\xi}}, \boldsymbol{\xi}) - \widetilde{L}_n^{(\mathrm{SKKM})}(\tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\xi}}) \le 2 \cdot \sup_{\substack{\boldsymbol{\eta}\in\{0,1\}^p \\ \boldsymbol{\nu}\in\mathcal{H}_k^{\boldsymbol{\eta}}}} \left|\widetilde{L}_n^{(\mathrm{SKKM})}(\boldsymbol{\nu}, \boldsymbol{\eta}) - L(\boldsymbol{\nu}, \boldsymbol{\eta})\right|.$$

Finally, by using Lemma 2.5, we have for any $\delta \in (2e^{-n}, 2e^{-1/n})$, there exist C_1, C_2 given in Lemma 2.5 such that

$$\Pr\left(\left|\widetilde{L}_n^{(\mathrm{SKKM})}(\tilde{\boldsymbol{\mu}}^{\boldsymbol{\xi}}, \boldsymbol{\xi}) - \widetilde{L}_n^{(\mathrm{SKKM})}(\tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\xi}})\right| \le \frac{2C_1 + 2C_2\sqrt{\log(4/\delta)}}{\sqrt{n}}\right) > 1 - \delta,$$

which completes the proof. □

2.7.7 Supplementary for Remark 2.2

In Remark 2.2, we claim that when d is exactly the true value d_0, or chosen by an empirical value $\tilde{d}_n$, the probability that the condition Eq. (2.11) holds converges to 1 for any λ_n with $\lim_{n\to\infty} \lambda_n = 0$ and $\lim_{n\to\infty} \lambda_n\sqrt{n} = \infty$. We here provide formal proofs for the two cases. Specifically, Proposition 2.3 is for the case of $d = d_0$, and the case of $d = \tilde{d}_n$ is the immediate result of Proposition 2.4, which shows that $\tilde{d}_n$ converges in probability to d_0.

Proposition 2.3. *Let $(\tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\xi}})$ be the maximizer of Eq. (2.7) with constraint $\|\boldsymbol{\xi}\|_0 \le d_0$, and for a fixed $\boldsymbol{\xi}$, we let $\tilde{\boldsymbol{\mu}}^{\boldsymbol{\xi}} = \arg\max_{\boldsymbol{\mu}\in\mathcal{H}_k^{\boldsymbol{\xi}}} \widetilde{L}_n^{(\mathrm{SKKM})}(\boldsymbol{\mu}, \boldsymbol{\xi})$, and*

$$\nabla_n^+(d_0) = \max_{\|\boldsymbol{\xi}\|_0 > d_0} \frac{\widetilde{L}_n^{(\mathrm{SKKM})}(\tilde{\boldsymbol{\mu}}^{\boldsymbol{\xi}}, \boldsymbol{\xi}) - \widetilde{L}_n^{(\mathrm{SKKM})}(\tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\xi}})}{\|\boldsymbol{\xi}\|_0 - d_0}$$
$$\nabla_n^-(d_0) = \min_{\|\boldsymbol{\xi}\|_0 < d_0} \frac{\widetilde{L}_n^{(\mathrm{SKKM})}(\tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\xi}}) - \widetilde{L}_n^{(\mathrm{SKKM})}(\tilde{\boldsymbol{\mu}}^{\boldsymbol{\xi}}, \boldsymbol{\xi})}{d_0 - \|\boldsymbol{\xi}\|_0}.$$

Under Conditions 2.1-2.4, for any λ_n with $\lim_{n\to\infty} \lambda_n = 0$ and $\lim_{n\to\infty} \lambda_n\sqrt{n} = \infty$, the following holds:

$$\lim_{n\to\infty} \Pr\left(\nabla_n^+(d_0) \le \lambda_n \le \nabla_n^-(d_0)\right) = 1.$$

Proof. At first, we give the following facts about $\nabla_n^+(d_0)$ and $\nabla_n^-(d_0)$:

(i) $\nabla_n^+(d_0) = O_P(1/\sqrt{n})$;

(ii) There exists a positive constant q such that

$$\lim_{n\to\infty} \Pr\left(\nabla_n^-(d_0) \ge \frac{q}{2d_0}\right) = 1.$$

Since (i) is given by Lemma 2.5, we only prove (ii). As a counterpart of $\widetilde{Q}_n(t)$, we define for any $t = 1, \ldots, p$,

$$Q(t) = \max\left\{L(\boldsymbol{\mu}, \boldsymbol{\xi}) \mid \boldsymbol{\xi} \in \{0,1\}^p, \|\boldsymbol{\xi}\|_0 \le t, \boldsymbol{\mu} \in \mathcal{H}_k^{\boldsymbol{\xi}}\right\}.$$

Then, by Condition 2.4, we have $Q(d_0 - 1) < Q(d_0)$, and then we can define

$$q = Q(d_0) - Q(d_0 - 1) > 0. \tag{2.31}$$

To prove (ii), we note that for a fixed $n \in \mathbb{N}_+$,

$$\begin{aligned}\nabla_n^-(d_0) &\geq \frac{1}{d_0} \cdot \min_{\|\boldsymbol{\xi}\|_0 < d_0} \left\{ \widetilde{L}_n^{(\mathrm{SKKM})}(\tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\xi}}) - \widetilde{L}_n^{(\mathrm{SKKM})}(\tilde{\boldsymbol{\mu}}^{\boldsymbol{\xi}}, \boldsymbol{\xi}) \right\} \\ &= \frac{1}{d_0} \cdot \left\{ \widetilde{L}_n^{(\mathrm{SKKM})}(\tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\xi}}) - \max_{\|\boldsymbol{\xi}\|_0 < d_0} \widetilde{L}_n^{(\mathrm{SKKM})}(\tilde{\boldsymbol{\mu}}^{\boldsymbol{\xi}}, \boldsymbol{\xi}) \right\} \\ &= \frac{1}{d_0} \cdot \left\{ \widetilde{Q}_n(d_0) - \max_{t < d_0} \widetilde{Q}_n(t) \right\},\end{aligned}$$

where the last equality is according to definitions $\widetilde{Q}_n(t)$. Moreover, since $\widetilde{Q}_n(t) \leq \widetilde{Q}_n(t+1)$ for any $t = 1, \ldots, p-1$, then we have

$$\nabla_n^-(d_0) \geq \frac{1}{d_0} \cdot \left\{ \widetilde{Q}_n(d_0) - \widetilde{Q}_n(d_0 - 1) \right\}. \tag{2.32}$$

Because

$$\begin{aligned}&\widetilde{Q}_n(d_0) - \widetilde{Q}_n(d_0 - 1) \\ &= \left[\widetilde{Q}_n(d_0) - Q(d_0)\right] + \left[Q(d_0) - Q(d_0 - 1)\right] + \left[Q(d_0 - 1) - \widetilde{Q}_n(d_0 - 1)\right] \\ &= \left[\widetilde{Q}_n(d_0) - Q(d_0)\right] + q + \left[Q(d_0 - 1) - \widetilde{Q}_n(d_0 - 1)\right],\end{aligned}$$

we turn to find lower bounds for the first and third terms, respectively. By the definition of $(\boldsymbol{\mu}^{**}, \boldsymbol{\xi}^{**})$ and writing $(\tilde{\boldsymbol{\nu}}, \tilde{\boldsymbol{\eta}})$ to be

$$(\tilde{\boldsymbol{\nu}}, \tilde{\boldsymbol{\eta}}) = \operatorname*{arg\,max}_{\|\boldsymbol{\xi}\|_0 \leq d_0 - 1,\ \boldsymbol{\mu} \in \mathcal{H}_k^{\boldsymbol{\xi}}} \widetilde{L}_n(\boldsymbol{\mu}, \boldsymbol{\xi}),$$

we have the followings hold:

$$\widetilde{Q}_n(d_0) - Q(d_0) = \widetilde{L}_n(\tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\xi}}) - \max_{\|\boldsymbol{\xi}\|_0 \leq d_0,\ \boldsymbol{\mu} \in \mathcal{H}_k^{\boldsymbol{\xi}}} L(\boldsymbol{\mu}, \boldsymbol{\xi}) \geq \widetilde{L}_n(\boldsymbol{\mu}^{**}, \boldsymbol{\xi}^{**}) - L(\boldsymbol{\mu}^{**}, \boldsymbol{\xi}^{**}),$$

and

$$Q(d_0 - 1) - \widetilde{Q}_n(d_0 - 1) = \max_{\|\boldsymbol{\xi}\|_0 \leq d_0 - 1,\ \boldsymbol{\mu} \in \mathcal{H}_k^{\boldsymbol{\xi}}} L(\boldsymbol{\mu}, \boldsymbol{\xi}) - \widetilde{L}_n(\tilde{\boldsymbol{\nu}}, \tilde{\boldsymbol{\eta}}) \geq L(\tilde{\boldsymbol{\nu}}, \tilde{\boldsymbol{\eta}}) - \widetilde{L}_n(\tilde{\boldsymbol{\nu}}, \tilde{\boldsymbol{\eta}}).$$

Then, by using Lemma 2.5, we have for any $\delta \in (2e^{-n}, 2e^{-1/n})$, there exists $M_\delta > 0$ such that

$$\begin{aligned}\Pr\left(\widetilde{L}_n(\boldsymbol{\mu}^{**}, \boldsymbol{\xi}^{**}) - L(\boldsymbol{\mu}^{**}, \boldsymbol{\xi}^{**}) \geq -\frac{M_\delta}{\sqrt{n}} \right) &> 1 - \frac{\delta}{2} \\ \text{and } \Pr\left(L(\tilde{\boldsymbol{\nu}}, \tilde{\boldsymbol{\eta}}) - \widetilde{L}_n(\tilde{\boldsymbol{\nu}}, \tilde{\boldsymbol{\eta}}) \geq -\frac{M_\delta}{\sqrt{n}} \right) &> 1 - \frac{\delta}{2}.\end{aligned}$$

It follows that

$$\Pr\left(\widetilde{Q}_n(d_0) - Q(d_0) \geq -\frac{M_\delta}{\sqrt{n}} \text{ and } Q(d_0-1) - \widetilde{Q}_n(d_0-1) \geq -\frac{M_\delta}{\sqrt{n}}\right) > 1-\delta.$$

Then we have

$$\Pr\left(\widetilde{Q}_n(d_0) - \widetilde{Q}_n(d_0-1) \geq q - \frac{2M_\delta}{\sqrt{n}}\right) > 1-\delta.$$

Consequently, for any $\delta \in (0,1)$, we take $N \in \mathbb{N}_+$ satisfying $2e^{-N} < \delta < 2e^{-1/N}$ and $2M_\delta/\sqrt{N} < q/2$, then for any $n \geq N$, we have $q - 2M_\delta/\sqrt{n} \geq q/2$, which follows that

$$\Pr\left(\widetilde{Q}_n(d_0) - \widetilde{Q}_n(d_0-1) \geq \frac{q}{2}\right) > 1-\delta.$$

Therefore, by combining Eq.(2.32), we complete the proof of (ii).

Second, based on the facts (i) and (ii), and the assumptions of λ_n, we prove the final result. For any $\delta \in (0,1)$, according to (i) and $\lim_{n\to\infty} \lambda_n\sqrt{n} = \infty$, we have there exists $M'_\delta > 0$ and $N_1 \in \mathbb{N}_+$ such that for any $n \geq N_1$,

$$\Pr\left(\nabla_n^+(d_0) \leq \frac{M'_\delta}{\sqrt{n}}\right) > 1-\frac{\delta}{2} \text{ and } \lambda_n > \frac{M'_\delta}{\sqrt{n}},$$

which means

$$\Pr\left(\nabla_n^+(d_0) \leq \lambda_n\right) > 1-\frac{\delta}{2}.$$

According to (ii) and $\lim_{n\to\infty} \lambda_n = 0$, we have there exists $N_2 \in \mathbb{N}_+$ such that for any $n \geq N_2$,

$$\Pr\left(\nabla_n^-(d_0) \geq \frac{q}{2d}\right) > 1-\frac{\delta}{2} \text{ and } \lambda_n < \frac{q}{2d},$$

which means

$$\Pr\left(\nabla_n^-(d_0) \geq \lambda_n\right) > 1-\frac{\delta}{2}.$$

Therefore, for any $n \geq \max\{N_1, N_2\}$, we have

$$\Pr\left(\nabla_n^+(d_0) > \lambda_n \text{ or } \nabla_n^-(d_0) < \lambda_n\right) \leq \delta,$$

which means

$$\Pr\left(\nabla_n^+(d_0) \leq \lambda_n \text{ and } \nabla_n^-(d_0) \geq \lambda_n\right) > 1-\delta.$$

We complete the proof. □

Proposition 2.4. *For $\tilde{d}_n$ defined by Eq.(2.12), that is,*

$$\tilde{d}_n = \min\{t \in \{1, \ldots, p\} \mid \widetilde{Q}_n(t) > \max_{s=1,\ldots,p} \widetilde{Q}_n(s) - \gamma_n\},$$

where $\lim_{n\to\infty} \gamma_n = 0$ *and* $\lim_{n\to\infty} \gamma_n\sqrt{n} = \infty$, *under Conditions 2.1-2.4, we have*

$$\lim_{n\to\infty} \Pr\left(\tilde{d}_n \neq d_0\right) = 0.$$

Proof. At first, we note that $\widetilde{Q}_n(t) \leq \widetilde{Q}_n(t+1)$ for any $t = 1, \ldots, p-1$, which implies that $\max_{s=1,\ldots,p} \widetilde{Q}_n(s) = \widetilde{Q}_n(p)$. We next turn to prove:

(i) $\lim_{n\to\infty} \Pr\left(\tilde{d}_n > d_0\right) = 0;$

(ii) $\lim_{n\to\infty} \Pr\left(\tilde{d}_n < d_0\right) = 0.$

(i) For any $n \in \mathbb{N}_+$, if $\tilde{d}_n > d_0$, then $\widetilde{Q}_n(d_0) \leq \max_{s=1,\ldots,p} \widetilde{Q}_n(s) - \gamma_n$, which means

$$\widetilde{Q}_n(p) - \widetilde{Q}_n(d_0) \geq \gamma_n.$$

According to the proof of Lemma 2.6, we have for any $\delta \in (0, 1)$, there exists $M_\delta > 0$ and $N_1 \in \mathbb{N}_+$ satisfying $2e^{-N_1} < \delta < 2e^{-1/N_1}$, such that for any $n \geq N_1$,

$$\Pr\left(\widetilde{Q}_n(p) - \widetilde{Q}_n(d_0) \leq \frac{M_\delta}{\sqrt{n}}\right) > 1 - \delta.$$

Moreover, because $\lim_{n\to\infty} \gamma_n\sqrt{n} = \infty$, then there exists $N_2 \in \mathbb{N}_+$ such that for any $n \geq N_2$, we have $\gamma_n > M_\delta/\sqrt{n}$. Consequently, for any $n \geq \max\{N_1, N_2\}$, we have

$$\Pr\left(\widetilde{Q}_n(p) - \widetilde{Q}_n(d) < \gamma_n\right) > 1 - \delta.$$

It follows that

$$\Pr\left(\tilde{d}_n > d_0\right) \leq \Pr\left(\widetilde{Q}_n(p) - \widetilde{Q}_n(d_0) \geq \gamma_n\right) \leq \delta,$$

which completes the proof of (i).

(ii) For any $n \in \mathbb{N}_+$, if $\tilde{d}_n < d_0$, then $\widetilde{Q}_n(\tilde{d}_n) \leq \widetilde{Q}_n(d_0)$. By the definition of $\tilde{d}_n$, we have

$$\max_{s=1,\ldots,p} \widetilde{Q}_n(s) \geq \widetilde{Q}_n(d_0) \geq \widetilde{Q}_n(\tilde{d}_n) > \max_{s=1,\ldots,p} \widetilde{Q}_n(s) - \gamma_n,$$

which means that

$$0 \leq \widetilde{Q}_n(d_0) - \widetilde{Q}_n(\tilde{d}_n) \leq \gamma_n.$$

On the other hand, $\tilde{d}_n < d_0$ implies $\tilde{d}_n \leq d_0 - 1$, which follows that $\widetilde{Q}_n(\tilde{d}_n) \leq \widetilde{Q}_n(d_0 - 1)$. Then we have

$$0 \leq \widetilde{Q}_n(d_0) - \widetilde{Q}_n(d_0 - 1) \leq \widetilde{Q}_n(d_0) - \widetilde{Q}_n(\tilde{d}_n) \leq \gamma_n.$$

According to the proof of Proposition 2.3 (the claim (ii)), we have for any $\delta \in (0, 1)$, there exists $N_1 \in \mathbb{N}_+$ such that for any $n \geq N_1$,

$$\Pr\left(\widetilde{Q}_n(d_0) - \widetilde{Q}_n(d_0 - 1) \geq \frac{q}{2}\right) > 1 - \delta.$$

Moreover, because $\lim\limits_{n\to\infty} \gamma_n = 0$, then there exists $N_2 \in \mathbb{N}_+$, such that for any $n \geq N_2$, we have $\gamma_n < q/2$. Consequently, for any $n \geq \max\{N_1, N_2\}$, we have

$$\Pr\left(\widetilde{Q}_n(d_0) - \widetilde{Q}_n(d_0 - 1) > \gamma_n\right) > 1 - \delta.$$

It follows that

$$\Pr\left(\tilde{d}_n < d_0\right) \leq \Pr\left(\widetilde{Q}_n(d_0) - \widetilde{Q}_n(d_0 - 1) \leq \gamma_n\right) \leq \delta,$$

which completes the proof of (ii). □

Chapter 3

Regularized k-POD clustering

3.1 Background

In this chapter, we focus on the situation when the data matrix is incomplete and includes missingness. The problem of missing data is ubiquitous in real-world applications for imperfect data collection processes. However, the issue of clustering for missing data, especially the k-means clustering for missing data receives far less attention. This problem occurs in many fields like astronomy (Wagstaff 2004, Almeida & Prieto 2013, Lithio & Maitra 2018), biology and medical science (Kiselev et al. 2017, Kim et al. 2019, Qi et al. 2020), where researchers need to divide plenty of astronomy signal or cells of patients into different groups, while the obtained original data is often incomplete or contaminated.

The main challenge is that the classical k-means clustering requires the data matrix to be complete, and thus directly conducting it on an incomplete data matrix is infeasible. The traditional approach is to pre-process the incomplete data matrix by complete-case analysis or multiple imputation to construct a new complete data matrix for conducting k-means clustering (Little & Rubin 2019). The complete-case analysis deletes all data points including missingness, which is also called the whole-data strategy in Hathaway & Bezdek (2001). If the missingness of each entry is completely at random, it is equivalent to conducting k-means clustering on a smaller dataset. However, when the missing proportion is large or the data dimension is high, there are few or even no such complete data points, and the complete-case analysis is no longer applicable in practice. Multiple imputation, such as *mice* (Van Buuren & Groothuis-Oudshoorn 2011) and *Amelia* (Honaker et al. 2011), predicts the missing entries of every data point based on its observed entries. It is equivalent to conducting k-means clustering on an approximated dataset. However, it relies on reasonable assumptions of data distributions and missingness mechanisms, which are of critical importance for accurate imputation (See Le Morvan et al. (2021) and Audigier & Niang (2023) for more discussion), and the computation cost is relatively high, especially for high-dimensional data. It should be noted that these pre-process steps are not only specific for the k-means clustering.

Alternatively, the k-POD clustering proposed by Chi et al. (2016) is a natural extension for k-means clustering to missing data and can be applicable for even large missingness proportions and high-dimensional data. However, the estimated cluster centers by k-POD clustering are generally biased, especially due to the existence of noise features in the high-dimensional space. This makes the corresponding clustering results unreliable. Therefore, our purpose in this chapter is to improve this method so that we can reduce the bias of estimated cluster centers and improve the performance of clustering.

3.1.1 Preliminaries for k-POD clustering

We first give some notations[1]. Write $\mathrm{X} = (x_{ij})_{n\times p} \in \mathbb{R}^{n\times p}$ for the data matrix with n data points $X_1, \dots, X_n$ in $\mathbb{R}^p$. Through this chapter, we encode the k cluster centers $\{\mu_1, \dots, \mu_k\}$ by a matrix $\mathrm{M} = (\mu_{lj})_{k\times p} \in \mathbb{R}^{k\times p}$, where the l-th row M_l represents the l-th cluster center μ_l. The membership between data points and cluster centers is denoted by a binary matrix $\mathrm{U} = (u_{il})_{n\times k} \in \{0,1\}^{n\times k}$, where $u_{il} = 1$ if and only if i-th data point X_i is assigned to l-th cluster. Since one data point is assigned to a unique cluster, it must satisfy that $\mathrm{U}\mathbf{1}_k = \mathbf{1}_n$, where $\mathbf{1}$ is the all-one vector. For a complete data matrix X, the k-means clustering can be expressed as

$$\min_{\mathrm{U},\mathrm{M}} \|\mathrm{X} - \mathrm{UM}\|_F^2, \tag{3.1}$$

where $\|\cdot\|_F$ is the Frobenius norm of a matrix, calculated as $(\sum_{i,j} a_{ij}^2)^{1/2}$ for $\mathrm{A} = (a_{ij})$. If there exist missing entries in X, the loss function cannot be directly calculated. Denoting all observed positions in X by a set $\Omega \subset \{1, \dots, n\} \times \{1, \dots, p\}$, the k-POD clustering introduces a mapping $\mathcal{P}$ onto the set Ω to replace the missing entries with zero. That is, $\mathcal{P}_\Omega : \mathbb{R}^{n\times p} \to \mathbb{R}^{n\times p}$, and $(\mathcal{P}_\Omega(\mathrm{X}))_{ij} = x_{ij}$ if $(i,j) \in \Omega$, 0 otherwise. Then, the k-POD clustering is given by

$$\min_{\mathrm{U},\mathrm{M}} \|\mathcal{P}_\Omega(\mathrm{X} - \mathrm{UM})\|_F^2. \tag{3.2}$$

[1] Through this chapter, we use $\|x\|$ to express l_2 norm of $x \in \mathbb{R}^p$.

The optimization procedure consists of filling in missing entries by the corresponding cluster means and conducting k-means clustering on the new data matrix alternatively. It should be noted that Wang et al. (2019) independently proposed the following:

$$\min_{\mathrm{Y},\mathrm{U},\mathrm{M}} \|\mathrm{Y}-\mathrm{UM}\|_F^2 \text{ such that } \mathrm{Y}\in\mathbb{R}^{n\times p}:\mathcal{P}_\Omega(\mathrm{Y})=\mathcal{P}_\Omega(\mathrm{X}). \tag{3.3}$$

Since the optimal solution of Eq. (3.3) should satisfy $\mathcal{P}_{\Omega^c}(\mathrm{Y})=\mathcal{P}_{\Omega^c}(\mathrm{UM})$, where Ω^c is the complement of Ω, it is actually identical to the solution of Eq. (3.2). Moreover, Lithio & Maitra (2018) proposed a variant of k-POD clustering, which substitutes the Lloyd's algorithm (Lloyd 1982) used in (Chi et al. 2016) by the Hartigan-Wong algorithm (Hartigan & Wong 1979) and shows comparable performance. Aschenbruck et al. (2023) proposed an adaptation of k-POD clustering based on substituting the k-means clustering to its extension k-prototypes algorithm, which is suitable for mixed type data with missingness.

However, the k-POD clustering is not consistent even under the missing completely at random mechanism (Terada & Guan 2024). The estimated cluster centers of the k-POD clustering and k-means clustering converge to different solutions as $n\to\infty$. The direct reason for the bias simply comes from the difference between loss functions of k-means and k-POD. Specifically, all positions of X are used by k-means, while only observed positions, i.e., $(i,j)\in\Omega$, are included by k-POD, and thus in general, one can hardly expect the same solutions based on these two different loss functions. We give two examples to illustrate this problem.

Example 3.1 ($p=2$)**.** *Suppose that the original complete data points (grey dots in Figure 3.1, $n=10^4$) are generated from a Gaussian mixture model in $\mathbb{R}^2$ with two equal components centered at $(0,2)$ and $(0,-2)$. When there is no missingness, the k-means clustering gives estimated cluster centers (black cross in left panel) being almost $(0,2)$ and $(0,-2)$. However, if all x_{ij} are missing completely at random with the missing probability being $2/3$ for x_{i1} and $1/3$ for x_{i2} and thus about 30% entries are missing, then the k-POD clustering on the incomplete data matrix gives estimated cluster centers (red triangles in central penal) by around $(0.7,2)$ and $(-0.7,-2)$, which are biased to the result of k-means clustering. Moreover, the resulting cluster boundary of k-POD clustering (red dotted line) is also skewed, compared with that of k-means clustering, which means a biased partition of the data space in the sense that even if a new data point is complete, it could be incorrectly classified.*

In Example 3.1, the bias of k-POD estimator occurs in the feature of the horizontal axis. It is actually a noise feature because there is no difference

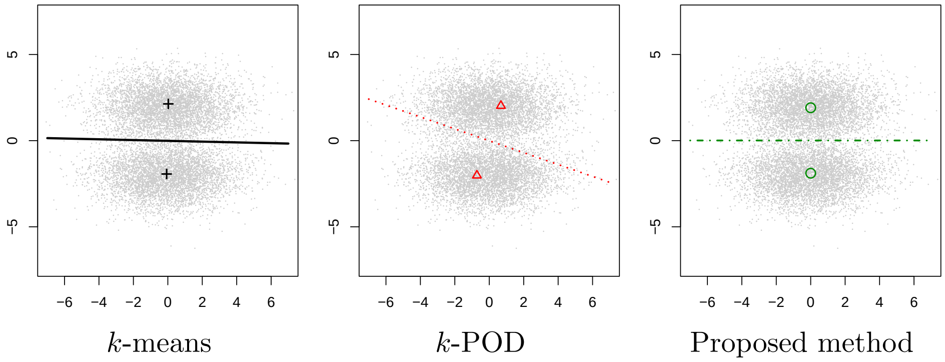


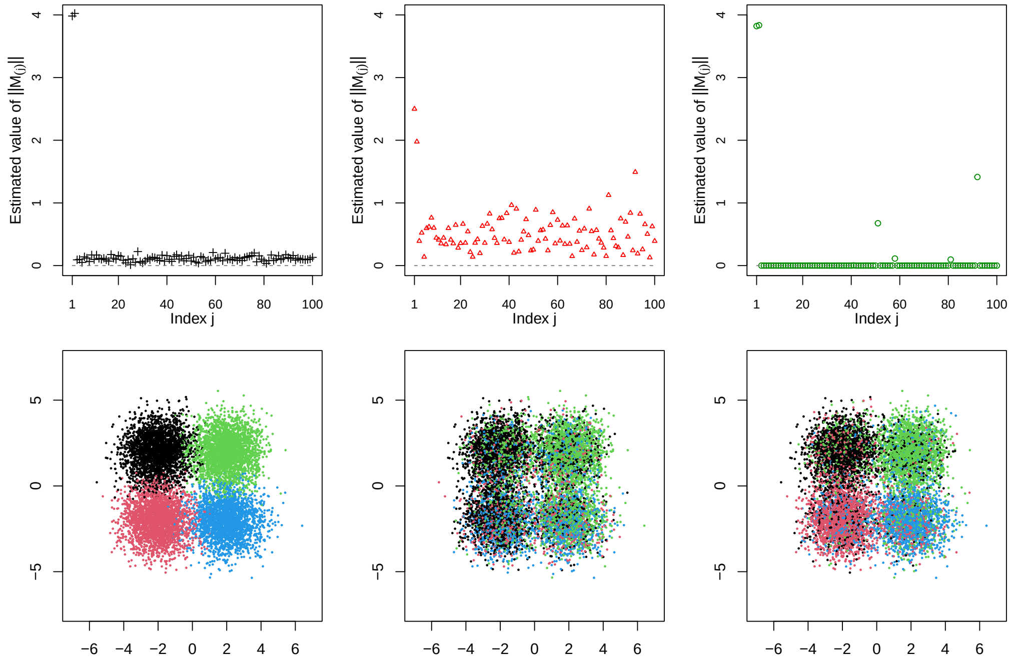
Figure 3.1: The estimated cluster centers of different methods for Example 3.1. The two axes represent two dimensions of data space. The grey dots are original complete data points. Colored points are estimated cluster centers. Lines are cluster boundaries associated with estimated cluster centers.

between the true cluster centers in this feature. This phenomenon also exists for the high-dimensional data, where it is common that only a few features are relevant to the true cluster structure and others are noise features.

Example 3.2 ($p = 100$)**.** *Consider the case of $p = 100$ with only the first two features being relevant to the true cluster structure. Suppose that the original complete data points generated from a Gaussian mixture model in $\mathbb{R}^{100}$ with four equal components centered at $(2, 2, 0, \ldots, 0)$, $(2, -2, 0, \ldots, 0)$, $(-2, 2, 0, \ldots, 0)$ and $(-2, -2, 0, \ldots, 0)$. Assume that each x_{ij} is missing completely at random with the missing probability being 0.3. We illustrate the l_2 norm of estimated cluster centers in each features of different methods (i.e., the norm of j-th column of M denoted by $\|\mathrm{M}_{(j)}\|$) in Figure 3.2 (top panels), and show the corresponding clustering results of different methods in Figure 3.2 (bottom panels), where only the first two features are used to illustrate data points. It can be seen that the k-POD estimators in noise features are significantly biased to zero, even though the k-means estimators (conducting on the no-missing data) are almost true. Moreover, the estimators in the first two features are also biased in this example. Therefore, the bias of estimated cluster centers of k-POD makes the corresponding clustering result almost fail.*

3.1.2 Our contribution

In this chapter, we propose regularized k-POD clustering for high-dimensional missing data. Specifically, we introduce a regularization function of cluster



k-means k-POD Proposed method

Figure 3.2: The results of Example 3.2. Top panels: The l_2 norm of estimated cluster centers in each features, i.e., $\|\mathrm{M}_{(j)}\|$ of different methods. The horizontal axis represents the index of feature j and the vertical axis is the estimated value of $\|\mathrm{M}_{(j)}\|$ $(j = 1, \ldots, p)$. Bottom panels: The clustering results of different methods. The two axes are the first two features. The color of each point represents the estimated cluster it is assigned to.

centers to the loss of k-POD clustering, which shrinks cluster centers featurewisely. This offers a significant advantage of reducing the bias of estimated cluster centers, in the case when noise features exist that have no contribution to the true cluster structure, which is common in high-dimensional space.

For the above two examples, since the bias of k-POD clustering occurs in the noise features, the shrinkage of estimated cluster centers in these features helps to reduce the bias. Specifically, for Example 3.1, the right panel of Figure 3.1 shows the result of the proposed method, where the estimated cluster centers (green circles) are closer to the result of k-means clustering (black cross in left panel) and the corresponding cluster boundary (green dashed line) is less skewed thus implies more reliable partition of data space. For Example 3.2, the right column of Figure 3.2 shows the result of the proposed method, where the estimated cluster centers are almost zero in noise features and are close to the true values in relevant features, and the corresponding clustering result is also more reliable.

In addition, we propose an optimization algorithm for the regularized k-POD clustering based on the majorization-minimization algorithm, which is an iteration between an imputation step and a clustering step. The experiments on synthetic datasets verify the reduction of bias and improvement of performance on clustering, and applications on real-world high-dimensional datasets also show better performance of the proposed method.

3.2 Proposed method

Suppose that the data matrix $\mathrm{X} = (x_{ij})_{n\times p}$ is column-wised centered, that is, $\frac{1}{n}\sum_{i=1}^{n} x_{ij} = 0$ for all $j = 1, \ldots, p$. The j-th column of X is denoted by $\mathrm{X}_{(j)} \in \mathbb{R}^n$ $(j = 1, \ldots, p)$. The set of observed positions of X is denoted by Ω. Suppose that the number of clusters $k \geq 2$ is fixed.

We define the loss function of regularized k-POD clustering with respect to membership $\mathrm{U} \in \{0, 1\}^{n\times k}$, $\mathrm{U}\mathbf{1}_k = \mathbf{1}_n$, and cluster centers $\mathrm{M} \in \mathbb{R}^{k\times p}$ to be

$$\widehat{L}_n(\mathrm{U}, \mathrm{M}) = \|\mathcal{P}_\Omega(\mathrm{X} - \mathrm{UM})\|_F^2 + \lambda \cdot J(\mathrm{M}). \tag{3.4}$$

The first term is the loss of the k-POD clustering, and $J(\mathrm{M})$ is a regularization function with respect to M. To shrink the estimated cluster centers feature-wisely, we consider two types of $J(\mathrm{M})$:

$$\text{The } l_0 \text{ penalty :} \quad J_0(\mathrm{M}) = \sum_{j=1}^{p} \mathbb{1}(\|\mathrm{M}_{(j)}\| > 0)$$

$$\text{The group lasso penalty :} \quad J_1(\mathrm{M}) = \sum_{j=1}^{p} w_j \|\mathrm{M}_{(j)}\|,$$

where $\mathrm{M}_{(j)} = (\mu_{1j}, \ldots, \mu_{kj})^T$ denotes the j-th column of cluster centers M with μ_{lj} being the j-th component of the l-th cluster center ($l = 1, \ldots, k$). The function $\mathbb{1}(\cdot)$ is the indicator function and w_j is the weight for $\mathrm{M}_{(j)}$. Both types of $J(\cdot)$ are column-wised, which means that all elements of $\mathrm{M}_{(j)}$, that is $\{\mu_{1j}, \ldots, \mu_{kj}\}$ would be shrunk together. The l_0 type $J_0(\cdot)$ constrains the number of non-zero columns of M, while the group lasso type $J_1(\cdot)$ constrains the weighted sum of l_2 norms of M in each feature. Therefore, with suitable regularization parameter λ, the estimated cluster centers $\widehat{\mathrm{M}}$ would be sparse in columns. The sparsity is further analyzed in Section 3.4.

In addition, the group lasso type contains weights. We note that in the framework of group lasso regression, a common choice for w_j is based on the square root of the size of j-th group (Yuan & Lin 2006, Yang & Zou 2015), which means a uniform weight $w_j = \sqrt{k}$ in our case. However, as in the above examples, the bias of the k-POD estimator in each feature is different, which implies that the adaptive weights are more reasonable. Specifically, in this paper, we consider the weights based on the k-POD estimator $\widetilde{\mathrm{M}}$, that is, $w_j = 1/\|\widetilde{\mathrm{M}}_{(j)}\|$. If the estimated cluster centers of the k-POD clustering in a feature are relatively concentrated, the corresponding weight would be relatively large, which makes the group lasso estimator in the corresponding feature more likely to be zero.

It should be noted that when the data matrix X is complete and $\Omega = \{1, \ldots, n\} \times \{1, \ldots, p\}$, the loss of the proposed method is equivalent to that of the regularized k-means clustering (Sun et al. 2012, Raymaekers & Zamar 2022). Therefore, the proposed method can also be viewed as an extension of the regularized k-means clustering to missing data.

3.3 Optimization

3.3.1 Algorithms

We apply the majorization-minimization algorithm (MM algorithm) (Hunter & Lange 2004) to minimize the proposed loss function Eq. (3.4). The MM algorithm constructs a majorization function $g(\theta \mid \theta^{(t)})$ for the original objective function $L(\theta)$ at the current value $\theta^{(t)}$, $t \in \mathbb{N}$. The majorization means that the domination condition $g(\theta \mid \theta^{(t)}) \geq L(\theta)$ and the tangency condition $g(\theta^{(t)} \mid \theta^{(t)}) = L(\theta^{(t)})$ are satisfied. Then update $\theta^{(t+1)}$ by minimizing $g(\theta \mid \theta^{(t)})$ instead of $L(\theta)$, which also guarantees $L(\theta^{(t+1)}) \leq L(\theta^{(t)})$.

Our goal is to minimize $\widehat{L}_n(\mathrm{U}, \mathrm{M})$ of Eq. (3.4) with respect to (U, M). We define the following function at current value $(\mathrm{U}^{(t)}, \mathrm{M}^{(t)})$, $t \in \mathbb{N}$:

$$g(\mathrm{U}, \mathrm{M} \mid \mathrm{U}^{(t)}, \mathrm{M}^{(t)}) = \|\mathcal{P}_{\Omega}(\mathrm{X} - \mathrm{UM})\|_F^2 + \lambda \cdot J(\mathrm{M}) + \|\mathcal{P}_{\Omega^c}(\mathrm{UM} - \mathrm{U}^{(t)}\mathrm{M}^{(t)})\|_F^2,$$

where Ω^c is the complement set of Ω. Because of the non-negativity of $\| \cdot \|_F^2$, the function $g(\mathrm{U}, \mathrm{M} \mid \mathrm{U}^{(t)}, \mathrm{M}^{(t)})$ is a majorization function of $\widehat{L}_n(\mathrm{U}, \mathrm{M})$ in the sense that

$$\begin{aligned} g(\mathrm{U}, \mathrm{M} \mid \mathrm{U}^{(t)}, \mathrm{M}^{(t)}) &\geq \widehat{L}_n(\mathrm{U}, \mathrm{M}) && \text{(domination condition)} \\ g(\mathrm{U}^{(t)}, \mathrm{M}^{(t)} \mid \mathrm{U}^{(t)}, \mathrm{M}^{(t)}) &= \widehat{L}_n(\mathrm{U}^{(t)}, \mathrm{M}^{(t)}) && \text{(tangency condition)} \end{aligned}$$

are both satisfied. If we use the notation $\widehat{\mathrm{X}} = \mathcal{P}_{\Omega}(\mathrm{X}) + \mathcal{P}_{\Omega^c}(\mathrm{U}^{(t)}\mathrm{M}^{(t)})$, then we have $g(\mathrm{U}, \mathrm{M} \mid \mathrm{U}^{(t)}, \mathrm{M}^{(t)}) = \|\widehat{\mathrm{X}} - \mathrm{UM}\|_F^2 + \lambda \cdot J(\mathrm{M})$. Notice that the matrix $\widehat{\mathrm{X}}$ is complete, then $g(\mathrm{U}, \mathrm{M} \mid \mathrm{U}^{(t)}, \mathrm{M}^{(t)})$ is actually the loss function of regularized k-means clustering on the data matrix $\widehat{\mathrm{X}}$. We then minimize the majorization function $g(\mathrm{U}, \mathrm{M} \mid \mathrm{U}^{(t)}, \mathrm{M}^{(t)})$ to update $(\mathrm{U}^{(t+1)}, \mathrm{M}^{(t+1)})$.

Therefore, we propose Algorithm 3.1 for regularized k-POD clustering. Specifically, given current $\mathrm{U}^{(t)}$ and $\mathrm{M}^{(t)}$, $t \in \mathbb{N}$, the $(t+1)$-th iteration consists of two steps. Step 1 imputes missing entries of X by the corresponding entries of multiplication matrix of current $\mathrm{U}^{(t)}$ and $\mathrm{M}^{(t)}$, so that we can get a new complete data matrix $\widehat{\mathrm{X}}^{(t+1)}$. Step 2 updates $\mathrm{U}^{(t+1)}$ and $\mathrm{M}^{(t+1)}$ by applying regularized k-means clustering on the imputed data matrix $\widehat{\mathrm{X}}^{(t+1)}$, the details of which is discussed later. Repeat the iteration until the loss (Eq. (3.4)) converges. Note that Algorithm 3.1 is a general framework for any type of $J(\cdot)$, and the difference in results comes from Step 2.

The convergence of Algorithm 3.1 to a local minima is guaranteed by the downhill trend

$$\widehat{L}_n(\mathrm{U}^{(t+1)}, \mathrm{M}^{(t+1)}) \leq \widehat{L}_n(\mathrm{U}^{(t)}, \mathrm{M}^{(t)})$$

for any $t \in \mathbb{N}$. This is the immediate consequence of the domination condition, tangency condition, and the definition of $(\mathrm{U}^{(t+1)}, \mathrm{M}^{(t+1)})$, which implies that

$$g(\mathrm{U}^{(t+1)}, \mathrm{M}^{(t+1)} \mid \mathrm{U}^{(t)}, \mathrm{M}^{(t)}) \leq g(\mathrm{U}^{(t)}, \mathrm{M}^{(t)} \mid \mathrm{U}^{(t)}, \mathrm{M}^{(t)}).$$

```
Algorithm 3.1 Regularized $k$-POD clustering
Input: incomplete data matrix $\mathrm{X}$, set of observed positions $\Omega$, number of clusters $k$.
Parameters: regularization parameter $\lambda$, weights $\{w_j\}$
  Initialize $\mathrm{U}^{(0)}$ and $\mathrm{M}^{(0)}$
  while Loss function (3.4) does not converge do
    1: Impute $\widehat{\mathrm{X}}^{(t+1)} = \mathcal{P}_\Omega(\mathrm{X}) + \mathcal{P}_{\Omega^c}(\mathrm{U}^{(\mathrm{t})}\mathrm{M}^{(\mathrm{t})})$
    2: Update $\mathrm{U}^{(t+1)}$ and $\mathrm{M}^{(t+1)}$ by applying Algorithm 3.2 on $\widehat{\mathrm{X}}^{(t+1)}$
  end while
Output: $\mathrm{U}^{(t+1)}$ and $\mathrm{M}^{(t+1)}$
```

According to our numerical experiments, the necessary number of iterations to convergence of the proposed method is generally comparable with that of the k-POD clustering.

Next, we introduce more details of Step 2 of Algorithm 3.1, where we apply regularized k-means clustering on imputed data matrix $\widehat{\mathrm{X}}^{(t+1)}$. For the simplification of notations, we here omit the superscript $(t+1)$ and focus on the general imputed complete data matrix $\widehat{\mathrm{X}}$. The goal of Step 2 of Algorithm 3.1 is to solve

$$\min_{\mathrm{U},\mathrm{M}} \|\widehat{\mathrm{X}} - \mathrm{U}\mathrm{M}\|_F^2 + \lambda \cdot J(\mathrm{M}), \tag{3.5}$$

with respect to $\mathrm{U} \in \{0,1\}^{n\times k}$, $\mathrm{U}\mathbf{1}_k = \mathbf{1}_n$ and $\mathrm{M} \in \mathbb{R}^{k\times p}$. Since it is not necessarily convex, an alternatively iterative procedure similar to Lloyd's algorithm (Lloyd 1982) for classical k-means clustering can be used. Therefore, we propose Algorithm 3.2 for this problem, which updates U and M separately. Specifically, given current $\mathrm{M}^{(r)}$, $r \in \mathbb{N}$, the membership $\mathrm{U}^{(r+1)}$ is determined by the distance between data points X_i and cluster centers $\mathrm{M}_l^{(r)}$, that is, $u_{il^*}^{(r+1)} = 1$ if $l^* = \arg\min_{1\le l\le k} \|\widehat{X}_i - \mathrm{M}_l^{(r)}\|^2$, 0 otherwise. Then, given $\mathrm{U}^{(r+1)}$, updating $\mathrm{M}^{(r+1)}$ depends on the types of $J(\cdot)$.

For $J = J_0$, the l_0 type, applying the KKT condition immediately leads to an explicit solution given by Eq. (3.8) that is a truncated version of the cluster means associated with current membership $\mathrm{U}^{(r+1)}$. For $J = J_1$, the group lasso type, since it is hard to derive an explicit expression, we apply the MM algorithm again to get $\mathrm{M}^{(r+1)}$. Denote by $f(\mathrm{M})$ the objective function in Eq. (3.5) with $\mathrm{U} = \mathrm{U}^{(r+1)}$ fixed and $J = J_1$, that is,

$$f(\mathrm{M}) = \|\widehat{\mathrm{X}} - \mathrm{U}^{(r+1)}\mathrm{M}\|_F^2 + \lambda \sum_{j=1}^{p} w_j \|\mathrm{M}_{(j)}\|. \tag{3.6}$$

At current $\mathrm{M}^{(r)}$, we define the following function:

$$h(\mathrm{M} \mid \mathrm{M}^{(r)}) = \|\widehat{\mathrm{X}} - \mathrm{U}^{(r+1)}\mathrm{M}\|_F^2 + \lambda \sum_{j=1}^{p} w_j \left(\frac{\|\mathrm{M}_{(j)}\|^2}{2\|\mathrm{M}_{(j)}^{(r)}\|} + \frac{1}{2}\|\mathrm{M}_{(j)}^{(r)}\| \right). \tag{3.7}$$

It can be proved that $h(\mathrm{M} \mid \mathrm{M}^{(r)})$ is a majorization of $f(\mathrm{M})$ at $\mathrm{M}^{(r)}$. Moreover, the solution of minimizing $h(\mathrm{M} \mid \mathrm{M}^{(r)})$ is explicit and given by Eq. (3.9), which can be viewed as a ridge version of the cluster means associated with the given membership $\mathrm{U}^{(r+1)}$. We thus use this solution as the update $\mathrm{M}^{(r+1)}$.

Algorithm 3.2 Regularized k-means clustering

```
Input: complete data matrix $\widehat{\mathrm{X}}$, number of clusters $k$.
Parameters: regularization parameter $\lambda$, weights $\{w_j\}$
  Initialize $\mathrm{M}^{(0)}$
  while Loss function (3.5) does not converge do
      a: Given $\mathrm{M}^{(r)}$, update $\mathrm{U}^{(r+1)}$ by: for any $i = 1, \dots, n$

         $u_{il^*}^{(r+1)} = \begin{cases} 1 & \text{if } l^* = \arg\min_{1 \le l \le k} \|\widehat{X}_i - \mathrm{M}_l^{(r)}\|^2 \\ 0 & \text{else} \end{cases}$

      b: Given $\mathrm{U}^{(r+1)}$, update $\mathrm{M}^{(r+1)}$ by: for any $j = 1, \dots, p$

      $(J = J_0) \quad \mathrm{M}_{(j)}^{(r+1)} = \begin{cases} \mathrm{V}_{(j)} & \text{if } \|\widehat{\mathrm{X}}_{(j)}\|^2 > \|\widehat{\mathrm{X}}_{(j)} - \mathrm{U}^{(r+1)}\mathrm{V}_{(j)}\|^2 + n\lambda \\ 0 & \text{else} \end{cases}$
                                                                                    (3.8)
             where $\mathrm{V}_{(j)} = \left(\mathrm{U}^{(r+1),T}\mathrm{U}^{(r+1)}\right)^{-1} \mathrm{U}^{(r+1),T}\widehat{\mathrm{X}}_{(j)}$

      $(J = J_1) \quad \mathrm{M}_{(j)}^{(r+1)} = \left( \mathrm{U}^{(r+1),T}\mathrm{U}^{(r+1)} + \frac{\lambda w_j}{2\|\mathrm{M}_{(j)}^{(r)}\|} \cdot \mathrm{I}_k \right)^{-1} \mathrm{U}^{(r+1),T}\widehat{\mathrm{X}}_{(j)}$   (3.9)

  end while
Output: $\mathrm{U}^{(r+1)}$ and $\mathrm{M}^{(r+1)}$
```

We give the following remarks for the update of $\mathrm{M}^{(r+1)}$ when $J = J_1$ and leave the technical details of Algorithm 3.2 in Section B.1 of Appendix B.

Remark 3.1. *The standard way to get $\mathrm{M}^{(r+1)}$ by MM algorithm is to do another iteration, that is, minimize $h(\mathrm{M} \mid \mathrm{M}^{(r_s)})$ on a sequence $\{\mathrm{M}^{(r_0)}, \mathrm{M}^{(r_1)}, \dots, \mathrm{M}^{(r_s)}\}$ about $s \in \mathbb{N}$ until convergence, which largely increases the computational cost.*

However, the multiple iteration for s is not necessary, since an update $\mathrm{M}^{(r+1)}$ *that reduces* $f(\mathrm{M})$ *is enough. Therefore, we can directly define* $h(\mathrm{M} \mid \mathrm{M}^{(r)})$ *based on current* $\mathrm{M}^{(r)}$*, and take the solution of minimizing* $h(\mathrm{M} \mid \mathrm{M}^{(r)})$ *to be the update* $\mathrm{M}^{(r+1)}$*. The optimality as well as majorization immediately implies* $f(\mathrm{M}^{(r+1)}) \leq f(\mathrm{M}^{(r)})$*. In this way, we can decrease the number of embedded loops and speed up the whole algorithm.*

Remark 3.2. *The minimization problem for* $f(\mathrm{M})$ *can be viewed as a group lasso regression of* $\widehat{\mathrm{X}}$ *on* $\mathrm{U}^{(r+1)}$*. Some existing literature that also considers MM algorithm uses the majorization based on a quadratic upper bound of* $\|\widehat{\mathrm{X}} - \mathrm{U}^{(r+1)}\mathrm{M}\|_F^2$ *(e.g.: Yang & Zou (2015)). Instead, we here use the upper bound of the penalty term* $\lambda \sum_{j=1}^{p} w_j \|\mathrm{M}_{(j)}\|$ *based on the basic inequality. According to comparisons provided in Appendix B, the performance of these two methods is quite similar. Refer to Section B.1.3 of Appendix B for more details.*

Finally, we analyze the computation complexity of the proposed algorithm. In Step 1 of Algorithm 3.1, imputing missing entries requires a complexity of $O(nkp + np(1-q))$, where q is the proportion of observed entries. In Step 2, updating U and M has the same complexity as the classical k-means clustering, i.e., $O(nkp\tau)$, where τ is the total number of iterations in Algorithm 3.2. Therefore, the asymptotic complexity of each iteration of the proposed algorithm is nearly $O(nkp\tau)$.

3.3.2 Initialization

Although the proposed algorithm has the guarantee to converge to some local minima, the multiple initialization should be considered, since the loss function of the proposed method is not necessarily convex with respect to U and M. In this paper, we consider two strategies to generate random initialization of $(\mathrm{U}^{(0)}, \mathrm{M}^{(0)})$.

The first strategy is based on the complete cases, which is referred to as `comp`. Specifically, we apply k-means++ clustering (Arthur & Vassilvitskii 2007) on the submatrix of X that only includes complete rows to obtain initial cluster centers $\mathrm{M}^{(0)}$. Then, the initial membership $\mathrm{U}^{(0)}$ is based on the Euclidean distances between data points and initial cluster centers. It should be noted that only the observed features are used to calculate the distance.

The second strategy is based on imputation, which is referred to as `impt`. Specifically, we pre-impute the incomplete data matrix X by column-wised sample means without considering missing entries. Then, we randomly sample k rows from the pre-imputed data matrix as the initial cluster centers

$\mathrm{M}^{(0)}$. The initial membership $\mathrm{U}^{(0)}$ is based on the Euclidean distances between data points and initial cluster centers. It should be noted that if there are duplicated rows in $\mathrm{M}^{(0)}$, some small noise is added to it to ensure k unique cluster centers.

Remark 3.3. *The two strategies use unique k random points to be initial k cluster centers and initialize membership based on them. According to our experiments, the empirical choice for the number of initialization is at least 100 to get more stable results. In the case of high-dimension or a large proportion of missingness, to reduce the computation cost, the sparse initialization (Raymaekers & Zamar 2022) can be used as an alternative. For example, based on the estimator by k-POD clustering, we can get several sparse submatrices of it by remaining columns with leading l_1 norms and letting others be zero, and then use these sparse submatrices to be initial cluster centers. Refer to Section B.3.2 of Appendix B for more details.*

3.3.3 Selection of tuning parameters

To select the tuning parameter, that is, the regularization parameter λ, we consider two kinds of criteria.

The first criterion is the instability of clustering (Wang 2010), which can be viewed as the cross-validation in the field of clustering. The main idea is that a good value for the tuning parameter should yield a stable clustering in response to minor disruption to the sample. The instability of a clustering algorithm $\boldsymbol{\psi}$ with tuning parameter λ is defined as

$$s(\boldsymbol{\psi};\lambda) = \mathbb{E}\left[D\left(\boldsymbol{\psi}(\mathrm{X}';\lambda), \boldsymbol{\psi}(\mathrm{X}'';\lambda)\right)\right],$$

where X' and X'' are two independent samples from the same distribution, and $\boldsymbol{\psi}(\mathrm{X}';\lambda)$ and $\boldsymbol{\psi}(\mathrm{X}'';\lambda)$ are two clustering trained on X' and X'', respectively. The notation $D(\cdot,\cdot)$ is the distance between two clusterings, which is given by the probability of the disagreement between them, that is,

$$D(\psi_1,\psi_2) = \mathbb{P}\left[\mathbb{1}(\psi_1(X) = \psi_1(\tilde{X})) + \mathbb{1}(\psi_2(X) = \psi_2(\tilde{X})) = 1\right],$$

where X and $\tilde{X}$ are two random variables independently sampled from the same distribution, ψ_1 and ψ_2 are two clusterings, and $\psi(x)$ indicates the cluster that x is assigned to.

In our case, the $\boldsymbol{\psi}$ is our clustering method that is based on cluster centers M, and by using the sample X' (or X'') and tuning parameter λ, the $\boldsymbol{\psi}(\mathrm{X}';\lambda)$ (or $\boldsymbol{\psi}(\mathrm{X}'';\lambda)$) is the estimated cluster centers $\widehat{\mathrm{M}}'$ (or $\widehat{\mathrm{M}}''$). The ψ_1 (or ψ_2) is the predicted cluster labels for some new data points based on $\widehat{\mathrm{M}}'$ (or $\widehat{\mathrm{M}}''$). The $D(\psi_1, \psi_2)$ is usually calculated by the disagreement of the two results of prediction. Specifically, for a given λ, the instability $s(\lambda)$ is calculated by repeating the following steps for B times, where the b-th repetition ($b = 1, \ldots, B$) consists of:

Step 1 Randomly divide the original sample X with sample size n into three subsets $\mathrm{X}', \mathrm{X}'', \tilde{\mathrm{X}}$, where X' and X'' have m data points, and $\tilde{\mathrm{X}}$ has $n - 2m$ data points;

Step 2 Conduct the proposed clustering method with λ on X' and X'' to obtain two estimators of cluster centers $\widehat{\mathrm{M}}'$ and $\widehat{\mathrm{M}}''$, respectively;

Step 3 Predict cluster labels for data points in $\tilde{\mathrm{X}}$, based on $\widehat{\mathrm{M}}'$ and $\widehat{\mathrm{M}}''$, respectively, and denote the two prediction results by ψ_1 and ψ_2;

Step 4 Calculate the disagreement between two prediction results $D(\psi_1, \psi_2)$ and denote it by D_b.

Finally, the instability is given by $s(\lambda) = \frac{1}{B}\sum_{b=1}^{B} D_b$. In addition, when the sample size n is small, the random division would make training sets too small. The bootstrap sampling can be an alternative to generate training and validation sets (Fang & Wang 2012).

The second criterion is the BIC index. Inspired by Raymaekers & Zamar (2022), Hofmeyr (2020), we use the following formulation:

$$\mathrm{BIC}(\lambda) = \|\mathcal{P}_\Omega(\mathrm{X} - \widehat{\mathrm{U}}\widehat{\mathrm{M}})\|_F^2 + \log(n) \cdot k \cdot d, \tag{3.10}$$

where $\widehat{\mathrm{U}}$ and $\widehat{\mathrm{M}}$ are estimators based on λ and $d = \sum_{j=1}^{p} \mathbb{1}(\|\widehat{\mathrm{M}}_{(j)}\| > 0)$ is the number of non-zero columns. The first term corresponds to the log-likelihood according to Fraley & Raftery (2002), while the second term is the degree of freedom, for which we use the number of independent parameters kd since the membership can be determined by cluster centers. We leave the technical details of derivation in Section B.2 of Appendix B.

For a set of values for λ, we select the best one with the smallest instability or BIC.

3.4 Theoretical properties

In this section, we further analyze some properties of the proposed method. We first introduce a binary matrix $\mathrm{R} = (r_{ij})_{n\times p} \in \{0,1\}^{n\times p}$ to indicate whether (i,j)-th entry is observed, that is, $r_{ij} = 1$ if (i,j) is observed, 0 otherwise. Then the incomplete data matrix can be expressed by $\mathrm{X} \circ \mathrm{R}$, where $\circ$ is the entry-wised multiplication. It is actually equivalent to the notation $\mathcal{P}_\Omega(\mathrm{X})$ of Eq. (3.4), because both of them substitute missingness by 0. The same as before that M_l denotes the l-th row of matrix M, we write R_i for the i-th row of matrix R. Recall that the i-th data point X_i is also the i-th row of data matrix X. Then the loss function of regularized k-POD clustering can be rewritten as

$$\widehat{L}_n(\mathrm{M}) = \sum_{i=1}^{n} \min_{l=1,\ldots,k} \|X_i \circ \mathrm{R}_i - \mathrm{M}_l \circ \mathrm{R}_i\|^2 + \lambda \cdot J(\mathrm{M}). \tag{3.11}$$

We note that this expression regards the loss as a function only with respect to M.

Write $\widehat{\mathrm{M}}$ for the minimizer of Eq. (3.11). Then, we can define the corresponding partition of the sample $\{X_1, \ldots, X_n\}$ in the following way. We first define a subset of the sample by

$$S_l = \{X_i \mid \|X_i \circ \mathrm{R}_i - \widehat{\mathrm{M}}_l \circ \mathrm{R}_i\| \leq \|X_i \circ \mathrm{R}_i - \widehat{\mathrm{M}}_{l'} \circ \mathrm{R}_i\|, \forall l' \neq l\}.$$

Since it is possible that $S_l \cap S_{l'} \neq \emptyset$ for some l, $l' \in \{1, \ldots, k\}$, then $\{S_1, \ldots, S_k\}$ is not a partition of $\{\mathrm{X}_i\}_{i=1}^n$. We instead define a sequence of subsets

$$\widehat{C}_l = S_l \setminus \left(\bigcup_{l'<l} S_{l'} \right).$$

Then $\widehat{\mathcal{C}} = \{\widehat{C}_1, \ldots, \widehat{C}_k\}$ forms a partition of the sample $\{X_1, \ldots, X_n\}$. Associated with $\widehat{\mathcal{C}}$, we define the corresponding membership matrix $\widehat{\mathrm{U}}$ by

$$\hat{u}_{il} = \mathbb{1}(X_i \in \widehat{C}_l), \quad \forall i = 1, \ldots, n, \quad l = 1, \ldots, k.$$

Write $\hat{q}_j$ for the proportion of observed entries in the j-th feature, and write $\bar{\mathrm{M}}_{(j)} = (\bar{\mu}_{1j}, \ldots, \bar{\mu}_{kj})^T$ and $\bar{\sigma}_j^2$ for the sample mean and variance in the j-th feature ignoring missing entries, that is,

$$\hat{q}_j = \frac{1}{n} \sum_{i=1}^{n} r_{ij} \quad \bar{\mu}_{lj} = \frac{1}{\sum_{i=1}^{n} \hat{u}_{il} r_{ij}} \sum_{i=1}^{n} \hat{u}_{il} r_{ij} x_{ij} \quad \bar{\sigma}_j^2 = \frac{1}{\sum_{i=1}^{n} r_{ij}} \sum_{i=1}^{n} r_{ij} x_{ij}^2.$$

Moreover, define the *within-cluster sum-of-square* associated with $\widehat{\mathcal{C}}$ in the j-th feature to be

$$\mathrm{WCSS}_j(\widehat{\mathcal{C}}) = \frac{1}{n} \sum_{i=1}^{n} \sum_{l=1}^{k} \mathbb{1}(X_i \in \widehat{C}_l) r_{ij} (x_{ij} - \bar{\mu}_{lj})^2.$$

Let $\widehat{Q}_j$ be the minimal value of the function Q_j with respect to $\mathrm{M}_{(j)}$, which is given by

$$Q_j(\mathrm{M}_{(j)}) = \frac{1}{n} \sum_{i=1}^{n} \min_{l=1,\ldots,k} r_{ij}(x_{ij} - \mu_{lj})^2.$$

The following proposition shows the sparsity of the estimated cluster centers $\widehat{\mathrm{M}}$ with different types of $J(\cdot)$, the proof of which is provided in Section 3.7.

Proposition 3.1. *(a) For $J(\cdot) = J_0(\cdot)$, if*

$$\hat{q}_j \bar{\sigma}_j^2 - WCSS_j(\widehat{\mathcal{C}}) \leq \frac{\lambda}{n},$$

then $\widehat{\mathrm{M}}_{(j)} = (0, 0, \ldots, 0)^T$. Otherwise, $\widehat{\mathrm{M}}_{(j)} \neq (0, 0, \ldots, 0)^T$ and has the l-th component $\hat{\mu}_{lj}$, $l = 1, \ldots, k$, satisfying:

$$\hat{\mu}_{lj} = \bar{\mu}_{lj}.$$

(b) For $J(\cdot) = J_1(\cdot)$ with weights $\{w_j\}_{j=1}^p$, if

$$\sqrt{\hat{q}_j \bar{\sigma}_j^2 - \widehat{Q}_j} < \frac{\lambda w_j}{2n},$$

then $\widehat{\mathrm{M}}_{(j)} = (0, 0, \ldots, 0)^T$. Otherwise, $\widehat{\mathrm{M}}_{(j)} \neq (0, 0, \ldots, 0)^T$ and has the l-th component $\hat{\mu}_{lj}$, $l = 1, \ldots, k$, satisfying:

$$\hat{\mu}_{lj} = \left(1 + \frac{\lambda w_j}{2 \cdot \|\widehat{\mathrm{M}}_{(j)}\| \cdot \sum_{i=1}^{n} \hat{u}_{il} r_{ij}}\right)^{-1} \cdot \bar{\mu}_{lj}.$$

Remark 3.4. *For $J = J_0$, those features in which the gap between total variance and WCSS is larger than a uniform threshold would be selected, and cluster centers in selected features are equal to the sample means. For $J = J_1$, the sparsity of cluster centers is determined by the weights, and cluster centers in selected features are a shrunk version of the sample means. Moreover, if there is no missing, this result coincides with that of regularized k-means clustering derived by Raymaekers & Zamar (2022) and Levrard (2018).*

3.5 Experiments

In this section, we empirically evaluate the performance of the proposed method. The incomplete datasets used in this section are constructed by artificially setting missing on original complete datasets. The structure of this section is as follows: (a) We describe the experimental setup in Section 3.5.1, including the generation of original complete data and the missingness mechanisms. (b) Focusing on the proposed method, we compare the effects of different strategies of initialization in Section 3.5.2. (c) We compare the effects of different criteria on the tuning parameter in Section 3.5.3. (d) The comparisons with other methods are summarized in Section 3.5.4.

3.5.1 Experimental setup

Complete data

For the original complete datasets, we consider synthetic datasets on which the k-means clustering performs well in the absence of missing data. The Gaussian mixture model of k components with equal weight $\frac{1}{k}$ and the same diagonal covariance matrix Σ is used, where the mean vector of l-th component is denoted by $\mu_l \in \mathbb{R}^p$, $l = 1, \ldots, k$. Specifically, the synthetic complete data points $X_i \in \mathbb{R}^p$, $i = 1, \ldots, n$, are generated as follows. For each i, we first uniformly sample z_i from $\{1, \ldots, k\}$ as the true cluster label. Then X_i is generated from a Gaussian distribution $\mathcal{N}(\mu_l, \Sigma)$ if $z_i = l$.

Through this section, we fix the sample size $n = 3000$ and the number of clusters $k = 4$, and the following μ_l's are used:

$$\begin{pmatrix} \mu_1^T \\ \mu_2^T \\ \mu_3^T \\ \mu_4^T \end{pmatrix} = \begin{pmatrix} a\mathbf{1}_{d/2}^T & a\mathbf{1}_{d/2}^T & \mathbf{0}_{p-d}^T \\ a\mathbf{1}_{d/2}^T & -a\mathbf{1}_{d/2}^T & \mathbf{0}_{p-d}^T \\ -a\mathbf{1}_{d/2}^T & a\mathbf{1}_{d/2}^T & \mathbf{0}_{p-d}^T \\ -a\mathbf{1}_{d/2}^T & -a\mathbf{1}_{d/2}^T & \mathbf{0}_{p-d}^T \end{pmatrix}.$$

Since each μ_l consists of d informative values and $p-d$ zeros and the covariance matrix is diagonal, for complete data matrix X, the first d features are relevant to cluster structure, while the other $p-d$ features are noise features. To make most peer methods applicable for comparison, through this section, we consider two cases of features:

- $p = 10$ and $d = 2$, where $a = 2$ and $\Sigma = \text{diag}(1, 1, 4, \ldots, 4)$
- $p = 100$ and $d = 10$, where $a = 1$ or $a = 0.8$ and $\Sigma = \text{diag}(1, \ldots, 1, 2, \ldots, 2)$.

Missingness mechanism

The mechanism of missingness is the cause of the missing values. There are three main types: missing completely at random (MCAR), missing at random (MAR), and missing not at random (MNAR) (Little & Rubin 2019). The

MCAR requires that the missingness of X is independent with X itself, and the MAR requires that the missingness is only dependent on the observed part of X. Otherwise, it is called MNAR. To match different missingness mechanisms, through this section, we consider four types of procedures for generating missingness and leave more details of settings in Section B.3.1 of Appendix B.

- MCAR: The missing probability is set to be a constant. For any $i = 1, \ldots, n$ and $j = 1, \ldots, p$,

  $$\mathbb{P}(x_{ij} \text{ is missing}) = \tau.$$

  Different τ is to meet the total proportion of missingness from 10% to 50%.

- MAR: We fix the first column of X to be observed and the missingness of the other $p-1$ columns is dependent on the first column. For any $i = 1, \ldots, n$ and $j = 2, \ldots, p$,

  $$\mathbb{P}(x_{ij} \text{ is missing}) = \frac{1}{1 + \exp(-\psi_1(x_{i1} - \psi_2))}.$$

  Different (ψ_1, ψ_2) are selected to meet the total proportion of missingness from 10% to 30%.

- MNAR1 (Self-masked (Sportisse et al. 2020)): The missing probability is determined by the value of the data itself. For any $i = 1, \ldots, n$ and $j = 1, \ldots, p$,

  $$\mathbb{P}(x_{ij} \text{ is missing}) = \frac{1}{1 + \exp(-\phi_1(x_{ij} - \phi_2))}.$$

  Different (ϕ_1, ϕ_2) are selected to meet the total proportion of missingness from 10% to 30%.

- MNAR2 (Chi et al. 2016): In each column of X, entries in the bottom 10%, 20% and 30% quantiles are set to be missing.

Evaluation indexes

Since we focus on the estimation of cluster centers, we use the mean-squared error (MSE) of the estimated cluster centers as the main index for evaluation.

Specifically, denote $\widehat{\mathrm{M}}$ to be the estimated cluster centers, and M^* to be the underlying true cluster centers. The MSE is defined as

$$\mathrm{MSE}(\widehat{\mathrm{M}}, \mathrm{M}^*) = \sum_{l=1}^{k} \min_{l'=1,\ldots,k} \|\widehat{\mathrm{M}}_l - \mathrm{M}^*_{l'}\|^2.$$

Since for the k-means clustering, M^* is defined by the minimizer of the loss function in the population level, it is often unknown. However, based on the consistency of the k-means clustering, we can substitute it with the estimator under a sufficiently large sample size. That is, we generate a complete dataset with sample size $N = 10^5$ following the same distribution as the original complete data, and apply the k-means clustering on it. The output cluster centers would be used as the substitute of M^*.

Moreover, to compare the performance of clustering, we use the classification error rate (CER) as the index for evaluation. Specifically, denote $\widehat{\mathrm{U}}$ to be the estimated membership matrix, of which the associated partition of data points is denoted by $\widehat{\mathcal{C}}$. Denote $\mathcal{C}^*$ to be the true partition of data points. The CER is defined as

$$\mathrm{CER}(\widehat{\mathcal{C}}, \mathcal{C}^*) = \frac{1}{\binom{n}{2}} \sum_{i>i'} \left| \mathbb{1}_{\widehat{\mathcal{C}}(i,i')} - \mathbb{1}_{\mathcal{C}^*(i,i')} \right|,$$

where $\mathbb{1}_{\mathcal{C}(i,i')} = 1$ if the i-th and i'-th data points are assigned to the same cluster according to the partition $\mathcal{C}$, 0 otherwise.

In addition, we further compare the influence of the estimated cluster centers on predicting the partition of a validation dataset. Specifically, we generate a validation dataset that is complete with sample size $n_0 = 400$ and follows the sample distribution as the original complete data, and calculate the partition of it based on the estimated cluster centers. We use the classification error rate of the predictive partition to the true partition as the index for evaluation, and we call it *predictive CER* for short.

3.5.2 Effects of different initialization strategies

For both the k-POD clustering and the proposed method, we consider two strategies for random initialization. One is based on complete data points (`comp` for short), while another is based on imputation (`impt` for short). Table 3.1 illustrates the averaged values of MSE (with standard deviation in bracket) of different methods using different initialization strategies. Here we only use the case of $p = 10$, since for $p = 100$, there is no complete data point left. It can be seen that the `impt` strategy generally performs better than the `comp` strategy for both k-POD clustering and the proposed method. Moreover, although the `comp` strategy can give smaller MSE for the proposed method when there is 10% missing, it becomes less effective when the missing proportion gets large because there are fewer available complete data points for initialization.

Table 3.1: MSE (standard deviation in brackets) using different strategies for random initialization

Missing mechanism	Missing proportion	k-POD		Reg. k-POD (group lasso)		Reg. k-POD (l_0)	
		`impt`	`comp`	`impt`	`comp`	`impt`	`comp`
MCAR	10%	1.994 (0.90)	2.454 (0.87)	0.118 (0.03)	0.094 (0.03)	0.038 (0.01)	0.025 (0.01)
	20%	6.419 (2.11)	10.598 (4.17)	0.872 (0.57)	3.401 (3.73)	0.079 (0.03)	0.460 (1.25)
	30%	16.665 (4.74)	21.647 (4.42)	1.853 (0.71)	8.943 (6.01)	0.097 (0.03)	5.830 (7.05)
	40%	26.030 (4.24)	30.941 (5.81)	3.160 (0.88)	12.480 (8.39)	1.139 (2.46)	18.454 (11.26)
MAR	10%	2.631 (1.05)	11.928 (4.66)	0.364 (0.24)	2.715 (5.81)	0.203 (0.05)	13.310 (3.79)
	20%	5.887 (1.83)	27.540 (4.80)	0.298 (0.07)	21.233 (8.44)	0.117 (0.04)	28.059 (4.35)
	30%	6.835 (1.90)	28.343 (5.49)	0.484 (0.31)	13.322 (11.76)	0.115 (0.03)	28.778 (5.42)
MNAR1	10%	5.959 (0.65)	6.260 (0.75)	1.151 (0.10)	1.083 (0.10)	0.462 (0.05)	0.637 (0.75)
	20%	15.740 (4.12)	17.191 (2.67)	3.932 (0.33)	3.706 (0.33)	0.283 (0.05)	8.979 (5.00)
	30%	21.314 (3.29)	24.917 (4.58)	2.301 (0.35)	4.797 (5.94)	0.210 (0.07)	9.252 (7.50)
MNAR2	10%	6.481 (0.39)	6.696 (0.46)	2.006 (0.12)	1.942 (0.11)	0.691 (0.07)	0.676 (0.07)
	20%	21.531 (1.02)	23.848 (2.03)	4.901 (0.24)	5.458 (1.31)	2.346 (0.15)	7.491 (4.78)
	30%	47.923 (3.07)	52.439 (5.00)	24.829 (0.44)	24.975 (0.72)	9.733 (4.99)	21.930 (8.02)

In addition, we found that the l_0 type of the proposed method is more sensitive to the initialization than the group lasso type. We thus need more random initialization points, which however increases computation cost. An alternative for random initialization is the sparse initialization, which has comparable performance and needs fewer initialization points. We provide more details in Section B.3.2 of Appendix B.

3.5.3 Selection of regularization parameter

In this section, we compare the instability and BIC criteria for selecting the regularization parameter. We take the case of $p = 100$, $d = 10$, and $a = 1$ as an example. We let the regularization parameter λ vary in a grid of 20 candidate values given by $10^{-3+(4s/19)}$ for $s = 0, 1, \ldots, 19$, and calculate the corresponding values of instability and BIC criteria. For the instability criterion, we use 30 repetitions of random division. Note that only the `impt` strategy of initialization is used here.

Table 3.2 illustrates the averaged values of MSE (with the averaged number of active features in brackets) based on the λ selected by BIC and instability. It can be seen that for both types of the proposed method, under MCAR and MAR mechanisms, the λ selected by instability gives smaller

MSE but larger/comparable number of active features than that selected by BIC. Under MNAR mechanisms, the instability criterion performs much better than the BIC criterion, especially for the l_0 type of proposed method. The main reason is that deriving the expression of BIC is based on the assumption that missingness is independent to the complete data. However, the instability follows the spirit of cross-validation and is defined by the clustering alignment. We provide more details of comparison in the case of $p = 10$ and how the regularization parameter influences the performance of the proposed method in Section B.3.3 of Appendix B.

Table 3.2: MSE (number of active features in brackets) of proposed method using different criteria for selecting λ

Missing mechanism	Missing proportion	Reg. k-POD (group lasso)		Reg. k-POD (l_0)	
		Instability	BIC	Instability	BIC
MCAR	10%	0.126 (47)	0.187 (14)	0.109 (10)	0.114 (10)
	20%	0.206 (29)	0.458 (11)	0.156 (10)	0.161 (10)
	30%	0.407 (29)	0.743 (12)	0.305 (10)	0.280 (10)
	40%	1.934 (15)	1.918 (16)	2.675 (13)	10.412 (12)
	50%	5.546 (20)	9.018 (13)	25.895 (22)	25.073 (23)
MAR	10%	0.150 (19)	0.175 (10)	0.131 (10)	0.152 (10)
	20%	0.140 (18)	0.182 (10)	0.126 (10)	0.434 (16)
	30%	0.204 (12)	0.228 (10)	0.166 (10)	0.164 (10)
MNAR1	10%	3.073 (98)	25.418 (100)	1.873 (10)	26.062 (100)
	20%	3.109 (77)	33.044 (100)	1.738 (10)	33.559 (100)
	30%	2.139 (85)	20.032 (100)	1.324 (10)	30.417 (100)
MNAR2	10%	4.696 (78)	29.490 (100)	2.693 (10)	31.177 (100)
	20%	40.286 (100)	96.354 (100)	99.507 (100)	99.540 (100)

3.5.4 Comparison with other methods

In this section, we compare the proposed method with other methods on synthetic incomplete datasets. We consider the following peer methods:

- Complete-case analysis. We delete all rows that includes missing and then apply the classical k-means clustering to estimate the cluster centers. It should be noted that we only report the result of this method for the case of $p = 10$ since there are almost no complete data points left in the case of $p = 100$.

- Multiple imputation. We impute the missing entries via the popular mice model (Van Buuren & Groothuis-Oudshoorn 2011). The R package mice is used to get several complete data matrices after imputation. Then we pool the imputed data using element-wise mean to combine the multiple imputations into a single dataset, on which the classical k-means clustering is used to estimate the cluster centers.

- The k-POD clustering. To compare the effects of different initialization strategies, we use a modified version of the original R package kpodclustr (Chi et al. 2016), and report the better result.

For both group lasso and l_0 types of the proposed method, we consider two strategies of random initialization (impt and comp) and two criteria for selecting λ (instability and BIC), and then report the best result.

We apply these methods on all synthetic incomplete datasets to estimate cluster centers M and membership matrix U, and then calculate the corresponding MSE, CER and predictive CER. Table 3.3, Table 3.4 and Table 3.5 are results of different methods on different synthetic incomplete datasets, respectively. We report the results of $a = 0.8$ for $p = 100$ here and leave that of $a = 1$ in Section B.3.4 of Appendix B for the sake of space. The reported values are averaged indexes of 30 repetitions with standard deviations in the brackets. The bold font indicates the best results.

It can be seen that the proposed method outperforms other methods for estimating cluster centers and clustering. Specifically, the l_0 type of proposed method performs better when p is small, the missingness proportion is small and the mechanism is simple. The group lasso type of proposed method is stable against large p, large missingness proportion and complicated mechanisms. The main reason is that the solution of the l_0 type is based on a truncated expression, while the solution of the group lasso type would adjust the selected features as well, which improves the performance even though the k-POD clustering performs poorly in some complicated cases.

It should be noted that in the case of $p = 100$ with MCAR mechanism and missingness proportion larger than 40%, the proposed method is less effective than the multiple imputation method Mice. It is because in this case, the MAR assumption of Mice is satisfied, and moreover, the relevant features are highly related, which makes the imputation of missing entries by Mice more accurate. Moreover, the MNAR2 mechanism is hard for all methods, which is because the missingness of each entry does not follow a probabilistic model and the reasonable imputation is more challenging.

Furthermore, we compare the computation time of different methods. Figure 3.3 illustrates the results in the case of $p = 100$ under MCAR mechanism with 30% missingness, MAR mechanism with 20% missingness, MNAR1 and MNAR2 mechanisms with 10% missingness. We can see that the computation time of the proposed method is comparable to that of the k-POD clustering. However, the multiple imputation method Mice costs significantly more time, which coincides with the results of Chi et al. (2016). In addition, the l_0 type of proposed method is more time-consuming than the group lasso type. It is because in Step b of Algorithm 3.2 with l_0 penalty, comparing the variance and the within-cluster sum-of-squares is needed, which costs more time.

Table 3.3: MSE (standard deviation in brackets) of different methods

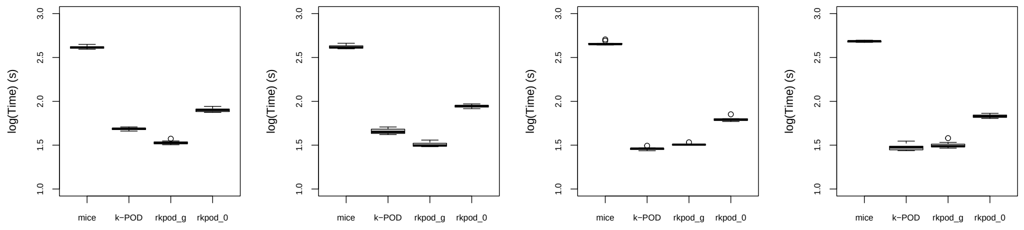
	Missing mechanism	Missing proportion	Complete-case analysis	Mice	k-POD	Reg. k-POD (group lasso)	Reg. k-POD (l_0)
$p = 10$	MCAR	10%	1.733 (1.15)	1.129 (0.75)	1.994 (0.90)	0.094 (0.03)	**0.025 (0.01)**
		20%	14.970 (5.08)	4.954 (2.24)	6.419 (2.11)	0.872 (0.57)	**0.079 (0.03)**
		30%	30.986 (5.36)	9.447 (2.30)	16.665 (4.74)	1.853 (0.71)	**0.097 (0.03)**
		40%	58.352 (12.80)	12.612 (2.23)	26.030 (4.24)	3.160 (0.88)	**1.139 (2.46)**
		50%	-	16.466 (2.20)	31.939 (5.47)	**4.732 (0.77)**	22.601 (6.93)
	MAR	10%	33.430 (1.13)	0.767 (0.23)	2.631 (1.05)	0.364 (0.24)	**0.203 (0.05)**
		20%	46.392 (1.60)	2.221 (1.53)	5.887 (1.83)	0.298 (0.07)	**0.117 (0.04)**
		30%	52.864 (5.71)	3.138 (1.98)	6.835 (1.90)	0.484 (0.31)	**0.115 (0.03)**
	MNAR1	10%	5.032 (0.76)	5.454 (0.85)	5.959 (0.65)	1.083 (0.10)	**0.462 (0.05)**
		20%	19.881 (3.59)	17.046 (1.39)	15.740 (4.12)	3.706 (0.33)	**0.283 (0.05)**
		30%	33.241 (6.39)	17.385 (1.50)	21.314 (3.29)	2.301 (0.35)	**0.210 (0.07)**
	MNAR2	10%	6.329 (0.67)	6.276 (0.33)	6.481 (0.39)	1.942 (0.11)	**0.676 (0.07)**
		20%	24.454 (2.49)	23.048 (2.41)	21.531 (1.02)	4.901 (0.24)	**2.356 (0.15)**
		30%	55.481 (7.27)	45.937 (1.78)	47.923 (3.07)	24.829 (0.44)	**9.733 (4.99)**
$p = 100$	MCAR	10%	-	1.916 (0.20)	2.558 (0.28)	0.153 (0.02)	**0.134 (0.02)**
		20%	-	2.239 (0.16)	4.612 (0.64)	0.162 (0.02)	**0.153 (0.03)**
		30%	-	2.768 (0.26)	15.475 (2.25)	**0.434 (0.10)**	7.948 (5.29)
		40%	-	**3.742 (0.45)**	25.168 (3.96)	6.938 (6.43)	26.469 (5.00)
		50%	-	**5.957 (0.63)**	36.216 (3.05)	23.472 (7.22)	36.284 (2.77)
	MAR	10%	-	1.948 (0.17)	2.483 (0.24)	0.197 (0.03)	**0.168 (0.04)**
		20%	-	2.181 (0.14)	6.130 (1.68)	0.246 (0.04)	**0.185 (0.03)**
		30%	-	2.657 (0.29)	11.834 (1.28)	**0.340 (0.10)**	6.495 (5.06)
	MNAR1	10%	-	26.022 (0.44)	26.514 (0.53)	**3.261 (0.14)**	4.963 (1.05)
		20%	-	33.406 (0.50)	35.853 (1.29)	**2.853 (0.19)**	6.562 (8.24)
		30%	-	26.842 (0.72)	39.057 (2.24)	**2.095 (0.31)**	40.053 (2.89)
	MNAR2	10%	-	32.759 (0.66)	33.161 (0.79)	**4.880 (0.18)**	16.871 (2.00)
		20%	-	104.249 (1.67)	109.296 (3.24)	**97.496 (2.95)**	109.614 (2.98)

3.6 Applications

In this section, we apply the proposed method to real-world datasets. We first consider the artificial missingness on some real-world complete datasets in Section 3.6.1. Then we evaluate the performance of the proposed method on real-world incomplete datasets in Section 3.6.2.

Table 3.4: CER (standard deviation in brackets) of different methods

	Missing mechanism	Missing proportion	Mice	k-POD	Reg. k-POD (group lasso)	Reg. k-POD (l_0)
$p = 10$	MCAR	10%	0.136 (0.01)	0.148 (0.02)	0.123 (0.01)	**0.123 (0.01)**
		20%	0.224 (0.02)	0.236 (0.02)	0.193 (0.01)	**0.186 (0.01)**
		30%	0.281 (0.01)	0.302 (0.01)	0.250 (0.01)	**0.241 (0.01)**
		40%	0.310 (0.01)	0.337 (0.01)	0.290 (0.01)	**0.285 (0.01)**
		50%	0.334 (0.00)	0.349 (0.01)	**0.315 (0.01)**	0.345 (0.01)
	MAR	10%	0.097 (0.01)	0.122 (0.01)	0.093 (0.01)	**0.090 (0.00)**
		20%	0.139 (0.01)	0.166 (0.01)	0.125 (0.00)	**0.124 (0.00)**
		30%	0.176 (0.01)	0.199 (0.01)	0.162 (0.01)	**0.161 (0.00)**
	MNAR1	10%	0.178 (0.01)	0.176 (0.02)	0.151 (0.01)	**0.149 (0.01)**
		20%	0.228 (0.00)	0.271 (0.02)	0.212 (0.01)	**0.202 (0.01)**
		30%	0.300 (0.00)	0.312 (0.01)	0.255 (0.01)	**0.251 (0.01)**
	MNAR2	10%	0.145 (0.00)	0.148 (0.01)	0.130 (0.00)	**0.130 (0.00)**
		20%	0.257 (0.02)	0.236 (0.02)	0.242 (0.01)	**0.210 (0.01)**
		30%	0.330 (0.00)	0.323 (0.01)	0.426 (0.03)	**0.292 (0.03)**
$p = 100$	MCAR	10%	0.109 (0.01)	0.118 (0.01)	0.094 (0.00)	**0.089 (0.01)**
		20%	0.135 (0.01)	0.175 (0.02)	**0.109 (0.01)**	0.113 (0.00)
		30%	0.165 (0.01)	0.288 (0.02)	**0.138 (0.00)**	0.245 (0.04)
		40%	**0.203 (0.01)**	0.357 (0.01)	0.248 (0.05)	0.375 (0.03)
		50%	**0.249 (0.01)**	0.376 (0.01)	0.359 (0.02)	0.376 (0.01)
	MAR	10%	0.109 (0.01)	0.118 (0.01)	**0.089 (0.01)**	0.092 (0.01)
		20%	0.131 (0.01)	0.192 (0.02)	**0.116 (0.00)**	0.122 (0.01)
		30%	0.161 (0.01)	0.257 (0.01)	**0.145 (0.01)**	0.229 (0.04)
	MNAR1	10%	0.129 (0.01)	0.132 (0.01)	**0.098 (0.00)**	0.104 (0.01)
		20%	0.150 (0.01)	0.190 (0.02)	**0.118 (0.01)**	0.176 (0.07)
		30%	0.175 (0.01)	0.300 (0.02)	**0.145 (0.01)**	0.311 (0.02)
	MNAR2	10%	0.149 (0.01)	0.158 (0.01)	**0.110 (0.01)**	0.136 (0.01)
		20%	**0.238 (0.01)**	0.294 (0.01)	0.304 (0.02)	0.313 (0.02)



MCAR (30%) MAR (20%) MNAR1 (10%) MNAR2 (10%)

Figure 3.3: The box plot of computation time of different methods in the case of $p = 100$. The group lasso and l_0 types of proposed method are denoted by *rkpod_g* and *rkpod_0* for short, respectively. The four panels from left to right are MCAR with 30% missingness, MAR with 20% missingness, MNAR1 and MNAR2 mechanisms with 10% missingness, respectively.

Table 3.5: Predictive CER (standard deviations in brackets) of different methods

	Missing mechanism	Missing proportion	Complete-case analysis	Mice	*k*-POD	Reg. *k*-POD (group lasso)	Reg. *k*-POD (l_0)
$p = 10$	MCAR	10%	0.075 (0.02)	0.065 (0.02)	0.081 (0.02)	0.042 (0.01)	**0.042 (0.01)**
		20%	0.201 (0.04)	0.131 (0.03)	0.142 (0.03)	0.060 (0.02)	**0.046 (0.01)**
		30%	0.278 (0.03)	0.183 (0.02)	0.218 (0.03)	0.075 (0.02)	**0.043 (0.01)**
		40%	0.335 (0.03)	0.210 (0.02)	0.267 (0.02)	0.102 (0.02)	**0.062 (0.04)**
		50%	-	0.240 (0.02)	0.284 (0.02)	**0.081 (0.03)**	0.249 (0.04)
	MAR	10%	0.074 (0.02)	0.076 (0.02)	0.094 (0.02)	0.048 (0.01)	**0.047 (0.01)**
		20%	0.220 (0.03)	0.220 (0.01)	0.197 (0.04)	0.046 (0.01)	**0.045 (0.01)**
		30%	0.274 (0.03)	0.228 (0.01)	0.240 (0.02)	0.050 (0.01)	**0.045 (0.01)**
	MNAR1	10%	0.074 (0.02)	0.076 (0.02)	0.094 (0.02)	0.048 (0.01)	**0.047 (0.01)**
		20%	0.220 (0.03)	0.220 (0.01)	0.197 (0.04)	0.046 (0.01)	**0.045 (0.01)**
		30%	0.274 (0.03)	0.228 (0.01)	0.240 (0.02)	0.050 (0.01)	**0.045 (0.01)**
	MNAR2	10%	0.065 (0.01)	0.060 (0.01)	0.073 (0.01)	0.048 (0.01)	**0.048 (0.01)**
		20%	0.175 (0.03)	0.136 (0.05)	0.128 (0.02)	**0.053 (0.01)**	0.060 (0.01)
		30%	0.323 (0.06)	0.203 (0.01)	0.247 (0.02)	0.248 (0.02)	**0.124 (0.05)**
$p = 100$	MCAR	10%	-	0.087 (0.01)	0.100 (0.01)	0.071 (0.01)	**0.071 (0.01)**
		20%	-	0.091 (0.01)	0.127 (0.02)	**0.072 (0.01)**	0.074 (0.01)
		30%	-	0.097 (0.01)	0.228 (0.02)	**0.066 (0.01)**	0.162 (0.06)
		40%	-	**0.112 (0.02)**	0.313 (0.02)	0.154 (0.08)	0.328 (0.04)
		50%	-	**0.142 (0.02)**	0.353 (0.01)	0.314 (0.04)	0.356 (0.01)
	MAR	10%	-	0.091 (0.01)	0.094 (0.01)	**0.068 (0.01)**	0.068 (0.01)
		20%	-	0.090 (0.01)	0.141 (0.02)	**0.069 (0.01)**	0.069 (0.02)
		30%	-	0.097 (0.01)	0.202 (0.02)	**0.070 (0.01)**	0.154 (0.06)
	MNAR1	10%	-	0.108 (0.01)	0.118 (0.01)	**0.080 (0.01)**	0.089 (0.01)
		20%	-	0.110 (0.01)	0.146 (0.03)	**0.079 (0.01)**	0.135 (0.09)
		30%	-	0.110 (0.01)	0.250 (0.02)	**0.083 (0.01)**	0.262 (0.03)
	MNAR2	10%	-	0.124 (0.01)	0.143 (0.01)	**0.089 (0.01)**	0.116 (0.02)
		20%	-	**0.231 (0.03)**	0.317 (0.03)	0.316 (0.04)	0.323 (0.03)

3.6.1 Real-world datasets with artificial missingness

Since we focus on the performance of estimating the cluster centers, the ground truth can be obtained only when the dataset includes no missingness. Therefore, we construct incomplete datasets in the same way as numerical experiments, that is, we artificially set missingness on original complete datasets.

We consider a microarray genomics dataset *Lymphoma*[2]. It consists of 4026 gene expressions ($p = 4026$), collected over 62 samples ($n = 62$). Out of the 62 samples, 42 are Diffuse Large B-Cell Lymphoma (DLBCL), 9 are Follicular Lymphoma (FL), and 11 are Chronic Lymphocytic Leukemia (CLL) cell samples ($k = 3$). The original dataset is complete and includes no missingness. We consider the MCAR mechanism with missing proportion from 10% to 50%, the MAR mechanism with missing proportion from 10% to 30%, and the MNAR1 and MNAR2 mechanisms with missing proportion from 10% to 20%. The generation of missingness for MCAR and MNAR mechanisms is similar to that of simulations. For the MAR mechanism, we fix the 40th feature to be complete, which is one of the most *influential* features according to analysis of existing literature, and the missingness of other features depends on the values of the 40th feature.

In this case, since p is much larger than n, there is no complete data point when artificial missingness is added, and the Complete-case analysis method is no longer applicable. Moreover, we cannot use the multiple imputation method, such as mice, because the computation time would be extremely long and not acceptable in practice. Therefore, we only compare the proposed method to the k-POD clustering.

To calculate the MSE for evaluation, the ground truth of cluster centers M^* is needed. According to existing literature (Sun et al. 2012, Jin & Wang 2016), for the *Lymphoma* dataset there exists a small subset of influential features, with which the k-means clustering can give a better clustering result. For example, the CER of classical k-means with all features is about 0.3, while that with 44 influential features is 0.05, which means that M^* is more likely to be sparse. Therefore, we use the result of Jin & Wang (2016) as an approximation of M^*.

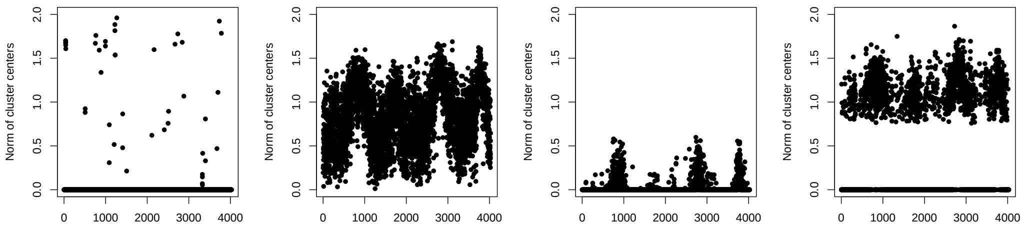
Table 3.6 illustrates the results of MSE for estimated cluster centers and CER for estimated membership of different methods. The reported values are the average of 10 repetitions with standard deviations in brackets. It can be seen that the proposed method, especially the group lasso type generally outperforms the k-POD clustering on both MSE and CER in various settings.

[2] The dataset can be found from https://www.stat.cmu.edu/~jiashun/Research/software/GenomicsData/Lymphoma/

Figure 3.4 illustrates the norm of estimated cluster centers in each feature for *Lymphoma* dataset under MCAR mechanism with 30% missing proportion. It can be seen that the results of the proposed method are more sparse than that of k-POD clustering. Moreover, since the l_0 type of proposed method is based on the hard threshold, there remain a lot of features, which leads to similar clustering performance to k-POD clustering. The group lasso type does not only select relevant features but also shrinks them, which leads to better performance.

Table 3.6: MSE and CER (standard deviations in brackets) of different methods for *Lymphoma* datasets

Missing mechanism	Missing proportion	MSE			CER		
		k-POD	Reg. k-POD (group lasso)	Reg. k-POD (l_0)	k-POD	Reg. k-POD (group lasso)	Reg. k-POD (l_0)
MCAR	10%	2077.987 (88.49)	**73.249 (0.54)**	1565.534 (139.37)	0.290 (0.01)	**0.135 (0.01)**	0.284 (0.01)
	20%	2193.763 (189.35)	**72.843 (0.34)**	1176.306 (246.66)	0.293 (0.01)	**0.130 (0.01)**	0.274 (0.06)
	30%	2254.447 (154.07)	**72.533 (0.06)**	1196.070 (175.43)	0.290 (0.01)	**0.123 (0.01)**	0.276 (0.08)
	40%	2299.094 (154.84)	**73.615 (0.49)**	1042.052 (193.14)	0.281 (0.02)	**0.145 (0.01)**	0.220 (0.12)
	50%	2448.054 (216.05)	**72.856 (0.27)**	828.744 (121.03)	0.308 (0.02)	**0.131 (0.01)**	0.180 (0.11)
MAR	10%	2092.131 (8.36)	**73.674 (0.31)**	1625.762 (165.52)	0.278 (0.00)	**0.156 (0.01)**	0.296 (0.01)
	20%	2182.586 (90.95)	**73.252 (0.25)**	1197.600 (254.64)	0.287 (0.01)	**0.157 (0.02)**	0.298 (0.01)
	30%	2284.245 (153.49)	**72.774 (0.13)**	1563.565 (281.55)	0.309 (0.02)	**0.177 (0.03)**	0.312 (0.02)
MNAR1	10%	1677.725 (7.37)	**75.813 (0.22)**	976.441 (108.01)	0.285 (0.00)	**0.163 (0.00)**	0.258 (0.08)
	20%	2209.57 (48.64)	**73.421 (0.59)**	1220.936 (157.73)	0.284 (0.01)	**0.136 (0.02)**	0.261 (0.08)
MNAR2	10%	1837.678 (0.00)	**73.456 (0.00)**	1050.854 (120.44)	0.300 (0.00)	**0.156 (0.00)**	0.274 (0.08)



ground truth

k-POD

Reg. k-POD (group lasso)

Reg. k-POD (l_0)

Figure 3.4: The estimated cluster centers of different methods for *Lymphoma* dataset under MCAR mechanism with 30% missing proportion. The x-axis is the feature index. The y-axis is the norm of cluster centers in each feature.

3.6.2 Real-world incomplete datasets

In this section, we evaluate the performance of the proposed method on real-world incomplete datasets. Since the ground truth of cluster centers of the complete dataset is unknown and cannot be approximated, we mainly concern with the practical effects of clustering.

We consider two single-cell RNA sequence datasets:

- *Usoskin* dataset contains 622 neuronal cells ($n = 622$) that are divided into four sensory subtypes ($k = 4$): peptidergic nociceptors, non-peptidergic nociceptors, neurofilament containing and tyrosine hydroxylase containing. We here use a subset of this dataset and corresponding labels provided by Usoskin et al. (2015), which consists of 452 genes ($p = 452$). The total missing proportion is about 73%.

- *Treutlein* dataset contains 265 cells ($n = 265$) that are in different states on the lineage from fibroblast to neuron, roughly including the initial MEF state, induced state, intermediate state, early and terminal neuron state. We here use a subset of this dataset and corresponding assignment of states provided by Treutlein et al. (2016), which consists of 396 genes ($p = 396$) and 7 types of states ($k = 7$). The total missing proportion is about 44%.

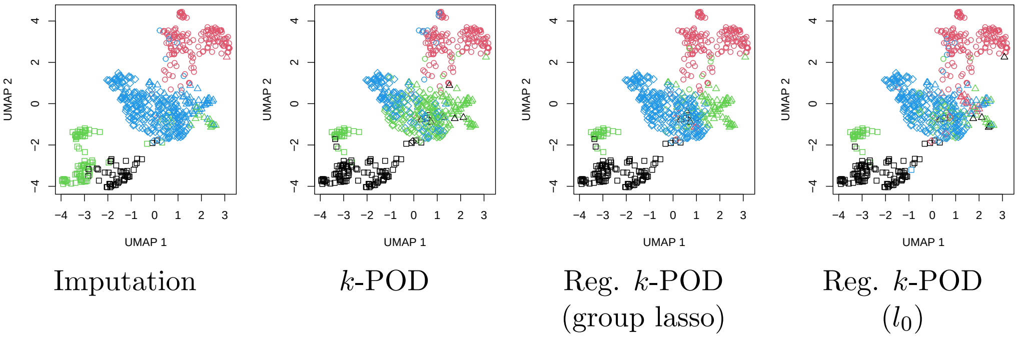
For both datasets, since p and the missing proportion are large, there is no complete data point left and thus the complete-case analysis method is no longer applicable. Moreover, the multiple imputation method takes extremely long time. Therefore, we consider the k-means clustering based on simple zero imputation and the k-POD clustering as peer methods for comparison.

Table 3.7 summarizes the averaged CER of 30 repetitions of different methods with standard deviation in brackets, and shows that the group lasso type of proposed method has the lowest CER and outperforms other methods on both datasets. This coincides with the results of numerical experiments, where the group lasso type of proposed method shows more stable and better performance in more complicated cases (large p and complicated missingness mechanism with a large proportion of missingness), because of the adjustment on both noise and relevant features.

In addition, for *Usoskin* dataset, we provide the visualization of clustering results in Figure 3.5 by using UMAP (Becht et al. 2019), where the shape of points represents the ground truth label and the color represents the estimated label. It shows that the group lasso type of proposed method gives a relatively more separated partition for 4 types of all 622 cells. For *Treutlein* dataset, we provide the estimated states from initial MEF to terminal neuron in Figure 3.6, where the y-axis represents the degree of identity of a cell to the terminal neuron state, and the x-axis represents the cell index ordered by the identity. The color of points represents the estimated state by different methods. It shows that the proposed method distinguishes states of the conversion more clearly, which corresponds to the rough distinction including the initial MEF state, induced state, intermediate state, early and terminal neuron state. However, other methods mix up the induced state and intermediate state.

Table 3.7: CER (standard deviations in brackets) of different methods for real-world incomplete datasets

Dataset	Imputation	k-POD	Reg. k-POD (group lasso)	Reg. k-POD (l_0)
Usoskin	0.138 (0.00)	0.198 (0.05)	**0.064 (0.01)**	0.167 (0.03)
Treutlein	0.110 (0.00)	0.126 (0.02)	**0.084 (0.01)**	0.136 (0.02)



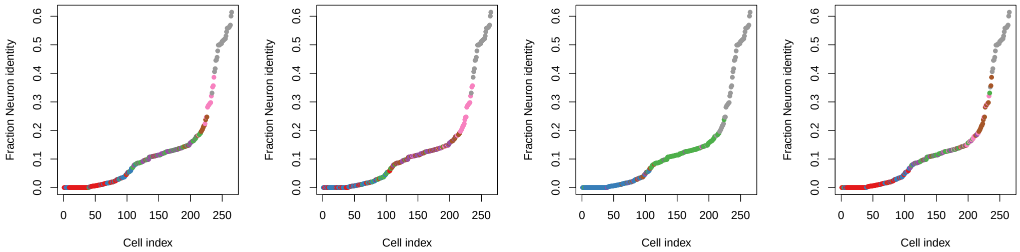
Imputation

k-POD

Reg. k-POD (group lasso)

Reg. k-POD (l_0)

Figure 3.5: The visualization of clustering results for cells in *Usoskin* dataset. The shape of points represents the true label and the color represents the estimated label.



Imputation

k-POD

Reg. k-POD (group lasso)

Reg. k-POD (l_0)

Figure 3.6: The conversion of estimated states for cells in *Treulein* datasets. The y-axis represents the identity of a cell to the terminal neuron state, and the x-axis represents the cell index ordered by the identity. The color of points represents the estimated state.

3.7 Proofs

In this section, we prove Proposition 3.1.

(a) For $J(\cdot) = J_0(\cdot)$, estimating $\widehat{\mathrm{M}}$ is equivalent to solving

$$\min_{\mathrm{M}_{(j)}} \sum_{i=1}^{n} r_{ij}(x_{ij} - \mathrm{U}_i \mathrm{M}_{(j)})^2 + \lambda \mathbb{1}(\|\mathrm{M}_{(j)}\| > 0)$$

for each $j = 1, \ldots, p$, where U is associated with M.

If the minimizer $\widehat{\mathrm{M}}_{(j)} \neq (0, 0, \ldots, 0)^T$, then $\mathbb{1}(\|\widehat{\mathrm{M}}_{(j)}\| > 0) = 1$ and the optimality according to KKT condition implies that

$$0 = -2 \sum_{i=1}^{n} r_{ij} \widehat{\mathrm{U}}_i^T (x_{ij} - \widehat{\mathrm{U}}_i \widehat{\mathrm{M}}_{(j)}).$$

It follows that for all $l = 1, \ldots, k$

$$\hat{\mu}_{lj} = \frac{\sum_{i=1}^{n} \hat{u}_{il} r_{ij} x_{ij}}{\sum_{i=1}^{n} \hat{u}_{il} r_{ij}} = \bar{\mu}_{lj}.$$

If $\widehat{\mathrm{M}}_{(j)} = (0, 0, \ldots, 0)^T$, then $\mathbb{1}(\|\widehat{\mathrm{M}}_{(j)}\| > 0) = 0$ and the optimality according to KKT condition implies that for any $\mathrm{V}_{(j)} \in \mathbb{R}^k$,

$$\sum_{i=1}^{n} r_{ij}(x_{ij} - \mathrm{U}_i^{(v)} \mathrm{V}_{(j)})^2 + \lambda \geq \sum_{i=1}^{n} r_{ij}(x_{ij} - \widehat{\mathrm{U}}_i \widehat{\mathrm{M}}_{(j)})^2,$$

where $\mathrm{U}^{(v)}$ is the membership matrix associated with V. Because

$$\sum_{i=1}^{n} r_{ij}(x_{ij} - \mathrm{U}_i^{(v)}\mathrm{V}_{(j)})^2 + \lambda \leq \sum_{i=1}^{n} r_{ij}(x_{ij} - \widehat{\mathrm{U}}_i\mathrm{V}_{(j)})^2 + \lambda$$

and

$$\sum_{i=1}^{n} r_{ij}(x_{ij} - \widehat{\mathrm{U}}_i\widehat{\mathrm{M}}_{(j)})^2 = \sum_{i=1}^{n} r_{ij}(x_{ij} - \widehat{\mathrm{U}}_i \cdot 0)^2 = \sum_{i=1}^{n} r_{ij}x_{ij}^2,$$

when we take $\mathrm{V}_{(j)} = \bar{\mathrm{M}}_{(j)} = (\bar{\mu}_{1j}, \ldots, \bar{\mu}_{kj})^T$, it follows that

$$\sum_{i=1}^{n} r_{ij}(x_{ij} - \widehat{\mathrm{U}}_i\bar{\mathrm{M}}_{(j)})^2 + \lambda \geq \sum_{i=1}^{n} r_{ij}x_{ij}^2.$$

Since the left hand is equivalent to $n \cdot \mathrm{WCSS}_j(\widehat{\mathcal{C}})$, we have

$$\lambda \geq n \cdot \hat{q}_j^2\bar{\sigma}_j^2 - n \cdot \mathrm{WCSS}_j(\widehat{\mathcal{C}}),$$

which completes the proof.

(b) For $J(\cdot) = J_1(\cdot)$ with weights $\{w_j\}_{j=1}^p$, estimating $\widehat{\mathrm{M}}$ is equivalent to solving

$$\min_{\mathrm{M}_{(j)}} \sum_{i=1}^{n} r_{ij}(x_{ij} - \mathrm{U}_i\mathrm{M}_{(j)})^2 + \lambda w_j \|\mathrm{M}_{(j)}\|$$

for each $j = 1, \ldots, p$.

If the minimizer $\widehat{\mathrm{M}}_{(j)} \neq 0$, then the optimality according to KKT condition implies that

$$\begin{aligned} 0 &= -2\sum_{i=1}^{n} r_{ij}\widehat{\mathrm{U}}_i^T(x_{ij} - \widehat{\mathrm{U}}_i\widehat{\mathrm{M}}_{(j)}) + \lambda w_j \frac{\widehat{\mathrm{M}}_{(j)}}{\|\widehat{\mathrm{M}}_{(j)}\|} \\ &= -2\widehat{\mathrm{U}}^T\left[\mathrm{X}_{(j)} \circ \mathrm{R}_{(j)} - (\widehat{\mathrm{U}}\widehat{\mathrm{M}}_{(j)}) \circ \mathrm{R}_{(j)}\right] + \lambda w_j \frac{\widehat{\mathrm{M}}_{(j)}}{\|\widehat{\mathrm{M}}_{(j)}\|}. \end{aligned}$$

It follows that

$$\left\|\widehat{\mathrm{U}}^T\left[\mathrm{X}_{(j)} \circ \mathrm{R}_{(j)} - (\widehat{\mathrm{U}}\widehat{\mathrm{M}}_{(j)}) \circ \mathrm{R}_{(j)}\right]\right\| = \frac{\lambda w_j}{2}.$$

Because $\widehat{\mathrm{U}}^T\left[\mathrm{X}_{(j)} \circ \mathrm{R}_{(j)} - (\widehat{\mathrm{U}}\widehat{\mathrm{M}}_{(j)}) \circ \mathrm{R}_{(j)}\right]$ is a vector in $\mathbb{R}^k$, the l-th component of which is

$$\begin{aligned}\sum_{i=1}^{n} \hat{u}_{il}\left[x_{ij}r_{ij} - (\widehat{\mathrm{U}}_i\widehat{\mathrm{M}}_{(j)}) \cdot r_{ij}\right] &= \sum_{X_i\in\widehat{C}_l}\left[x_{ij}r_{ij} - (\widehat{\mathrm{U}}_i\widehat{\mathrm{M}}_{(j)}) \cdot r_{ij}\right] \\ &= \sum_{X_i\in\widehat{C}_l}\left(x_{ij}r_{ij} - \hat{\mu}_{lj}r_{ij}\right),\end{aligned}$$

and

$$\begin{aligned}\left[\sum_{X_i\in\widehat{C}_l}\left(x_{ij}r_{ij} - \hat{\mu}_{lj}r_{ij}\right)\right]^2 &= \left[\sum_{X_i\in\widehat{C}_l} x_{ij}r_{ij} - \left(\sum_{X_i\in\widehat{C}_l} r_{ij}\right)\cdot\hat{\mu}_{lj}\right]^2 \\ &= \left(\sum_{X_i\in\widehat{C}_l} r_{ij}\right)^2 \cdot \left[\frac{\sum_{X_i\in\widehat{C}_l} x_{ij}r_{ij}}{\sum_{X_i\in\widehat{C}_l} r_{ij}} - \hat{\mu}_{lj}\right]^2,\end{aligned}$$

then

$$\frac{1}{n^2}\left\|\widehat{\mathrm{U}}^T\left[\mathrm{X}_{(j)} \circ \mathrm{R}_{(j)} - (\widehat{\mathrm{U}}\widehat{\mathrm{M}}_{(j)}) \circ \mathrm{R}_{(j)}\right]\right\|^2 = \sum_{l=1}^{k}\left(\frac{1}{n}\sum_{X_i\in\widehat{C}_l} r_{ij}\right)^2 \cdot \left[\frac{\sum_{X_i\in\widehat{C}_l} x_{ij}r_{ij}}{\sum_{X_i\in\widehat{C}_l} r_{ij}} - \hat{\mu}_{lj}\right]^2.$$

Since $r_{ij} \in \{0,1\}$ and $\widehat{C}_l \subset \{X_i\}_{i=1}^n$, then $\sum_{X_i\in\widehat{C}_l} r_{ij} \le n$, and it follows that $\left(\frac{1}{n}\sum_{X_i\in\widehat{C}_l} r_{ij}\right)^2 \le \frac{1}{n}\sum_{X_i\in\widehat{C}_l} r_{ij}$. Moreover, denote $\bar{\mu}_{lj} = \frac{\sum_{X_i\in\widehat{C}_l} x_{ij}r_{ij}}{\sum_{X_i\in\widehat{C}_l} r_{ij}}$, then we have

$$
\begin{aligned}
&\sum_{l=1}^{k}\left(\frac{1}{n}\sum_{X_i\in\widehat{C}_l} r_{ij}\right)\cdot(\bar{\mu}_{lj}-\hat{\mu}_{lj})^2\\
&=\frac{1}{n}\sum_{l=1}^{k}\left(\sum_{X_i\in\widehat{C}_l} r_{ij}\right)\cdot\left(\bar{\mu}_{lj}^2+\hat{\mu}_{lj}^2-2\bar{\mu}_{lj}\hat{\mu}_{lj}\right)\\
&=\frac{1}{n}\sum_{l=1}^{k}\left(\sum_{i=1}^{n}\mathbb{1}(X_i\in\widehat{C}_l, r_{ij}=1)\right)\cdot\left(\hat{\mu}_{lj}^2-\bar{\mu}_{lj}^2-2\bar{\mu}_{lj}\hat{\mu}_{lj}+2\bar{\mu}_{lj}^2\right)\\
&=\frac{1}{n}\sum_{i=1}^{n}\sum_{l=1}^{k}\mathbb{1}(X_i\in\widehat{C}_l, r_{ij}=1)\hat{\mu}_{lj}^2-\frac{1}{n}\sum_{i=1}^{n}\sum_{l=1}^{k}\mathbb{1}(X_i\in\widehat{C}_l, r_{ij}=1)\bar{\mu}_{lj}^2\\
&\quad-\frac{2}{n}\sum_{i=1}^{n}\sum_{l=1}^{k}\mathbb{1}(X_i\in\widehat{C}_l, r_{ij}=1)\hat{\mu}_{lj}x_{ij}+\frac{2}{n}\sum_{i=1}^{n}\sum_{l=1}^{k}\mathbb{1}(X_i\in\widehat{C}_l, r_{ij}=1)\bar{\mu}_{lj}x_{ij}\\
&=\frac{1}{n}\sum_{i=1}^{n}\sum_{l=1}^{k}\mathbb{1}(X_i\in\widehat{C}_l, r_{ij}=1)\left[(x_{ij}-\hat{\mu}_{lj})^2-(x_{ij}-\bar{\mu}_{lj})^2\right]\\
&=\frac{1}{n}\sum_{i=1}^{n}\sum_{l=1}^{k}\mathbb{1}(X_i\in\widehat{C}_l)r_{ij}\left(x_{ij}-\hat{\mu}_{lj}\right)^2-\frac{1}{n}\sum_{i=1}^{n}\sum_{l=1}^{k}\mathbb{1}(X_i\in\widehat{C}_l)r_{ij}\left(x_{ij}-\bar{\mu}_{lj}\right)^2.
\end{aligned}
$$

We next bound the two parts. Denote $\widehat{\mathrm{V}}=(\nu_{lj})_{k\times p}\in\mathbb{R}^{k\times p}$ to be the sparse modification of $\widehat{\mathrm{M}}$ with j-th column being zero, that is, $\widehat{\mathrm{V}}_{(j)}=0$ and $\widehat{\mathrm{V}}_{(j')}=\widehat{\mathrm{M}}_{(j')}$ for any $j'\neq j$. Because $\widehat{\mathrm{M}}$ minimizes $\widehat{L}_n(\mathrm{M})$ and the partition $\widehat{\mathcal{C}}=\{\widehat{C}_1,\ldots,\widehat{C}_k\}$ is determined by $\widehat{\mathrm{M}}$, it follows that

$$
\begin{aligned}
\widehat{L}_n(\widehat{\mathrm{M}})&=\sum_{i=1}^{n}\min_{l=1,\ldots,k}\|X_i\circ\mathrm{R}_i-\widehat{\mathrm{M}}_l\circ\mathrm{R}_i\|^2+\lambda\cdot J_1(\widehat{\mathrm{M}})\\
&=\sum_{i=1}^{n}\sum_{l=1}^{k}\mathbb{1}(X_i\in\widehat{C}_l)\|X_i\circ\mathrm{R}_i-\widehat{\mathrm{M}}_l\circ\mathrm{R}_i\|^2+\lambda\cdot J_1(\widehat{\mathrm{M}})\\
&\leq\sum_{i=1}^{n}\sum_{l=1}^{k}\mathbb{1}(X_i\in\widehat{C}_l)\|X_i\circ\mathrm{R}_i-\widehat{\mathrm{V}}_l\circ\mathrm{R}_i\|^2+\lambda\cdot J_1(\widehat{\mathrm{M}}).
\end{aligned}
$$

Considering the definition of group lasso penalty, that is, $J_1(\mathrm{M})=\sum_{j'=1}^{p}w_{j'}\|\mathrm{M}_{(j')}\|$, we thus have $J_1(\widehat{\mathrm{V}})\leq J_1(\widehat{\mathrm{M}})$, because $\widehat{\mathrm{V}}$ equals to $\widehat{\mathrm{M}}$ expect for j-th column.

It follows that

$$\sum_{i=1}^{n}\sum_{l=1}^{k}\mathbb{1}(X_i \in \widehat{C}_l)\|X_i \circ \mathrm{R}_i - \widehat{\mathrm{M}}_l \circ \mathrm{R}_i\|^2 \leq \sum_{i=1}^{n}\sum_{l=1}^{k}\mathbb{1}(X_i \in \widehat{C}_l)\|X_i \circ \mathrm{R}_i - \widehat{\mathrm{V}}_l \circ \mathrm{R}_i\|^2$$

and in the j-th column,

$$\begin{aligned}\sum_{i=1}^{n}\sum_{l=1}^{k}\mathbb{1}(X_i \in \widehat{C}_l)(x_{ij}r_{ij} - \hat{\mu}_{lj}r_{ij})^2 &\leq \sum_{i=1}^{n}\sum_{l=1}^{k}\mathbb{1}(X_i \in \widehat{C}_l)(x_{ij}r_{ij} - \hat{\nu}_{lj}r_{ij})^2\\ &= \sum_{i=1}^{n}\sum_{l=1}^{k}\mathbb{1}(X_i \in \widehat{C}_l)(x_{ij}r_{ij} - 0)^2\\ &= \sum_{i=1}^{n} r_{ij}x_{ij}^2 = n\hat{q}_j\bar{\sigma}_j^2.\end{aligned}$$

Therefore, the first term is bounded by $\hat{q}_j\bar{\sigma}_j^2$.

For the second part, we have

$$\frac{1}{n}\sum_{i=1}^{n}\sum_{l=1}^{k}\mathbb{1}(X_i \in \widehat{C}_l)r_{ij}\left(x_{ij} - \bar{\mu}_{lj}\right)^2 \geq \frac{1}{n}\sum_{i=1}^{n}\min_{l=1,\ldots,k} r_{ij}(x_{ij} - \bar{\mu}_{lj})^2.$$

The right hand must be larger than the minimal value of function Q_j, which implies that

$$\frac{1}{n}\sum_{i=1}^{n}\sum_{l=1}^{k}\mathbb{1}(X_i \in \widehat{C}_l)r_{ij}\left(x_{ij} - \bar{\mu}_{lj}\right)^2 \geq \widehat{Q}_j.$$

Combining the above all, we have

$$\frac{1}{n^2}\left\|\widehat{\mathrm{U}}^T\left[\mathrm{X}_{(j)} \circ \mathrm{R}_{(j)} - (\widehat{\mathrm{U}}\widehat{\mathrm{M}}_{(j)}) \circ \mathrm{R}_{(j)}\right]\right\|^2 \leq \hat{q}_j\bar{\sigma}_j^2 - \widehat{Q}_j,$$

which implies that

$$\frac{\lambda w_j}{2n} \leq \sqrt{\hat{q}_j\hat{\sigma}_j^2 - \widehat{Q}_j}.$$

Therefore, if for any j,

$$\frac{\lambda w_j}{2n} > \sqrt{\hat{q}_j\bar{\sigma}_j^2 - \widehat{Q}_j},$$

then $\widehat{\mathrm{M}}_{(j)} = 0$, which completes the proof.

Chapter 4

Discussions and future works

4.1 Discussions

In this thesis, we focused on improving the classical k-means clustering for complex high-dimensional data, especially the data with non-linear cluster structure and missing values. To this end, we proposed two novel clustering methods based on k-means clustering. The superior performance of the proposed methods verified the effectiveness of clustering high dimensional data with non-linear cluster structure and missing values. As a consequence, we make the traditional k-means clustering applicable for more complex data.

In Chapter 2, we proposed a novel sparse kernel k-means clustering to extend the advantages of kernel k-means clustering to the high-dimensional cases. The numerical experiments on synthetic and real-world datasets and statistical analysis verified the practical efficacy and theoretical soundness of the proposed method. We further illustrated the successful application of the proposed method to the normalized cut.

There are still some limitations of the sparse kernel k-means clustering. (1) The selection procedure of the proposed algorithm is based on simple forward selection, which is vulnerable to the local solution and heavily relies on a good initialization. (2) The implementation of the proposed method considered the number of clusters k being fixed and the kernel function being Gaussian kernel. Whereas in practice, we often have little information about the true cluster structure and need further efforts to determine an appropriate k, and we also need to select suitable kernel functions for different field data. In addition, as for the effects of potential influence factors, we only discussed the bandwidth of the kernel function ν^2 and the number of selected features d, while others still remain unclear at present. (3) Since the high computational cost of kernel-based methods is a well-known drawback, relatively, the proposed method is slower than other linear clustering methods.

In Chapter 3, we proposed the regularized k-POD clustering to reduce the bias of the existing k-POD clustering method for high-dimensional missing data. According to numerical experiments and real-world data applications, we found that when there exist noise features that have no contribution to cluster structure, the proposed method has less bias of estimated cluster centers, which helps to give a more reasonable result of clustering than other methods.

The main limitations of regularized k-POD clustering are as follows. (1) For regularization functions, we only considered the l_0 penalty and group lasso penalty. Whereas, the study of Raymaekers & Zamar (2022) shows that common l_1 and l_2 penalties also have effects on shrinking cluster centers for complete data. Whether these penalties work well for missing data still remains unclear at present. (2) To tune the regularization parameter, we considered the instability and BIC criteria, where we derived the Eq. (3.10) as the counterpart of the traditional formulation of BIC for missing data. However, the likelihood part of Eq. (3.10) requires the MCAR mechanism, and according to Hofmeyr (2020), the degree of freedom in clustering needs more precise approximation than the simple number of independent parameters. This makes the derived BIC perform poorly for some complex missingness mechanisms. (3) Since the proposed method is aimed at high-dimensional missing data with noise features, the effects of reducing bias would be poor when the sparsity structure is not satisfied. In addition, the performance of the proposed method is related to that of k-POD clustering. When the missing proportion is large, the failure of k-POD clustering would lead to poor results of the proposed method.

4.2 Future works

Due to the limitations of the proposed methods discussed above, the future works following this thesis will focus on improving proposed methods, and extending them to other related methods and fields of clustering for data with non-linear cluster structure and missing values.

For data with non-linear cluster structure, we would like to improve the current sparse k-means clustering method in two respects. (1) To get a better local solution, we will consider effective initialization strategies, such as the k-means++ (Arthur & Vassilvitskii 2007), which initializes k cluster centers one by one by sampling them from the original data points. The probability of each point being sampled is proportional to the nearest distance of it to the current cluster centers. In our case, the distance will be measured in the RKHS $\mathcal{H}$, and based on the initialized cluster centers, we can obtain the initialized partition. (2) To speed up the proposed algorithm, since the computational cost mainly comes from the construction of kernel matrix H, then it would be helpful to use the low-rank approximation of kernel matrix, especially when the sample size n is large. We will consider the Nyström method (Williams & Seeger 2000), which gives an approximation of H by calculating some blocks involving only a small subset of the sample. In addition, when the data dimension is high, it is also helpful to speed up, if we immigrate the obvious noise features by using the K-S testing before applying the proposed method.

For missing data, the regularized k-POD clustering needs further improvement and investigations as follows. (1) To make the BIC criteria more applicable for various missing data, it is necessary to derive formulations similar to Eq. (3.10) for other missingness mechanisms. To do so, we will first consider some specific MNAR mechanisms for data following the mixture distribution, such as the missingness only depending on the cluster membership. According to Sportisse et al. (2024), by concatenating the data matrix with the missing mask, the inference problem for model-based clustering under MNAR can be transformed into that under MAR. We will consider the similar idea to modify the likelihood part of BIC formulation. Also, the precise approximation of the degree of freedom proposed by Hofmeyr (2020) will be considered as well. (2) We would like to study the behavior of the proposed method in the population level. Since the k-POD clustering is a natural extension of k-means clustering to missing data, the proposed method can be viewed as the counterpart of regularized k-means clustering in the case of missingness. Moreover, the theoretical analysis for the limited sample and population of the regularized k-means clustering has been provided by Levrard (2018), and we found that their results for the limited sample coincide with our results in Section 3.4, when there is no missingness. Therefore, we will apply their framework and techniques to provide statistical guarantees for the proposed method.

Finally, there are some potential and interesting topics for future work. One topic is to apply our analysis to provide statistical guarantees for general k-means clustering methods, since the consistency of feature selection is one of the most important issues for high-dimensional data clustering. Another topic is to study the properties of k-POD clustering for the limited sample and figure out the necessary conditions under which the bias of k-POD clustering can be ignored.

Appendix A

Appendix for Chapter 2

A.1 Discussion on Algorithm 2.2

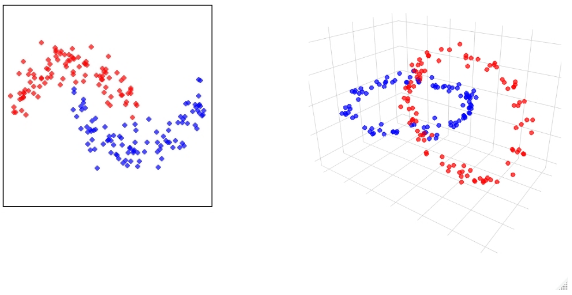
In this section, we provide some experimental discussion on the proposed algorithm for SKKM.

A.1.1 The number of iterations to convergence

The number of iterations to convergence is generally related to the complexity of clustering a dataset. The immediate convergence may be the common characteristic of the alternative algorithms for the simultaneous clustering and feature selection issue.

We in this section discuss the influence of possible factors, in addition to Figure 2.2, which illustrates the canonical trend of convergence. Specifically, we consider two synthetic datasets that consist of low-dimensional relevant features contributing to the true cluster structure, and several noise features. Figure A.1 illustrates the true cluster structure in the low dimensional space. We note that roughly speaking, the two moons dataset is easier to be clustered while the chainlink dataset is more hard. Then we compare the numbers of iterations to convergence of these two datasets with different sample sizes (n), different proportions of relevant features (d/p) and different kernel functions. The results are summarized in Table A.1. The reported values are the average numbers of iterations to convergence among 20 repetitions, as well as the standard deviations.

It can be seen that for each dataset, there is no significant difference between different sample sizes, different proportions of relevant features and different kernel functions. However, the difference between these two datasets is relatively significant. Since these two datasets roughly represent different difficulties of clustering problems, the limited comparison result implies that the number of iterations to convergence may increase when the clustering problem itself is difficult. According to this conclusion, we could explain that for the examples in Figure 2.2, the middle panel *Colon* dataset is relatively complicated even as a supervised classification task, which thus needs more iterations to convergence compared with other two examples.



Two moons Chainlink

Figure A.1: The underlying true cluster structures in the low dimensional space.

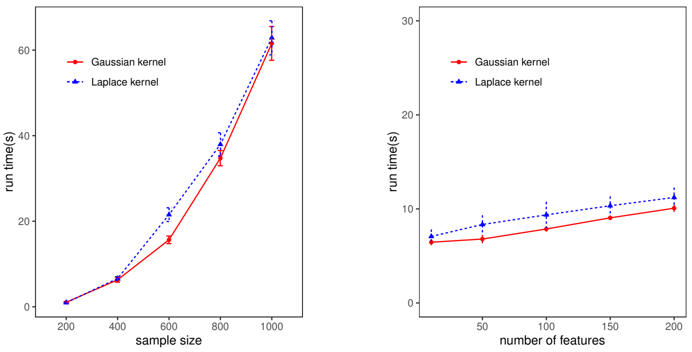
Table A.1: The comparison of the number of iterations to convergence

Dataset	n		d/p		Kernel function	
	200	1000	0.3	0.1	Gaussian	Laplacian
Two moons	3 (0)	3 (0)	3 (0)	3 (0)	3 (0)	3 (0)
Chainlink	5.7 (2.5)	4.7 (2.1)	5.7 (2.5)	5.3 (2.4)	5.7 (2.5)	3.5 (0.7)

A.1.2 The run time to convergence

In this section, we discuss the run time, in addition to the above experiments of the number of iterations. Specifically, we compare the run time under different sample sizes, different numbers of dimensions and different kernel functions. The results are illustrated in Figure A.2. The reported values are the average run time and corresponding error bars of 20 receptions.

It can be seen that sample size and number of features significantly have an influence on the run time while the kernel function being used shows little effects. Moreover, the trend of left panel about sample size is quadratic-liked, while right panel about the number of features is a linear trend. Therefore,



Sample size Number of features

Figure A.2: The effects of sample size, number of features and kernel functions on the run time.

the experiment results are in good agreement with our complexity analysis, which implies that the run time in practice may heavily rely on the size of the dataset. In addition, according to the run time on real datasets in Table A.2, a similar result is also observed.

Table A.2: Comparison of run time (seconds) for the real-world datasets

Dataset	$n \times p$	IF-PCA	Sparse k-means	Sparse MinMax k-means	**Proposed**
Glass	214 × 9	0.08	0.06	1.73	0.15
Breast	699 × 9	0.12	0.12	1.39	0.48
Vehicle	846 × 18	0.23	0.25	3.99	3.23
Trace	200 × 275	0.73	0.35	2.14	39.27
Control	600 × 60	0.50	0.40	6.39	19.28
Brain	42 × 5597	2.05	2.06	10.40	31.58
Colon	62 × 2000	0.90	0.57	1.32	10.73
Leukemia	72 × 3571	1.61	1.27	2.28	26.84
Lymphoma	62 × 4026	1.63	0.58	3.44	46.61
SRBCT	63 × 2308	1.07	0.91	3.51	10.68

A.2 Settings of synthetic datasets in Section 2.6.3

In Section 2.6.3, we discuss the performance of feature selection of the proposed method (SKKM) based on some synthetic datasets. All synthetic datasets consist of low-dimensional ground truth cluster structure and high-dimensional noise features. The details are summarised in Table A.3. We use the R package `mlbench`[1] to generate the ground truth distribution. For data1 to data4, we generate Gaussian mixture distribution on $\mathbb{R}^3$ by the function `mlbench.simplex` with arguments $n = 200$, $d = 3$. For data5 to data8, we generate a shape distribution on $\mathbb{R}^2$ called *Smiley* by the function `mlbench.smiley` with arguments $n = 200, sd1 = 0.2, sd2 = 0.2$.

Table A.3: Details of synthetic datasets used in Section 2.6.3

Dataset	Ground truth	Noise feature	$n \times p$
data1	Gaussian mixture	independent Normal distribution	200×13
data2	Gaussian mixture	correlated Normal distribution	200×13
data3	Gaussian mixture	independent Normal distribution	200×103
data4	Gaussian mixture	independent $\chi^2(5)$ distribution	200×103
data5	Smiley	independent Normal distribution	200×12
data6	Smiley	correlated Normal distribution	200×12
data7	Smiley	independent Normal distribution	200×102
data8	Smiley	independent $\chi^2(5)$ distribution	200×102

A.3 More details and results of weighted version (SWKKM)

In this section, we introduce more details and experimental results about the weighted version of the proposed method (SWKKM) in Section 2.6.6.

A.3.1 The construction of weights

As stated in Section 2.6.6, we mainly focus on the specific type of weighted kernel k means that coincides with normalized cut (Ncut). The corresponding weights and kernel function are given as follows. Denote the kernel function used in Ncut by $g : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. For the sample $\{x_1, \dots, x_n\}$, we define the empirical degree function of g by $\hat{d}_n : \mathcal{X} \to \mathbb{R}$ with

$$\hat{d}_n(x) = \frac{1}{n} \sum_{i'=1}^{n} g(x, x_{i'}).$$

When we let weights w_i for each data point x_i be

$$w_i = \hat{d}_n(x_i),$$

[1] https://cran.r-project.org/web/packages/mlbench/

and use the kernel function $\tilde{h}_n : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ with the form of

$$\tilde{h}_n(x, y) = \frac{g(x, y)}{\hat{d}_n(x)\hat{d}_n(y)},$$

then the weighted kernel k-means clustering using $\{w_i\}$ and $\tilde{h}_n$ coincides with Ncut using kernel g. Based on this fact, in our experiments, we take the function g to be Gaussian kernel and apply the corresponding $\{w_i\}$ and $\tilde{h}_n$ to the proposed method (Eq. (2.5)). In other words, we successfully apply the proposed method (SWKKM) to the normalized cut.

A.3.2 Experiments on synthetic datasets

In this section, we further evaluate the performance of both clustering and feature selection of the proposed method (SWKKM) on synthetic datasets. Ten artificial datasets are used, all consisting the ground truth relevant features and noise features. The decision boundaries between the ground truth clusters are all non-linearly separable. Similarly, the normal distribution $\mathcal{N}(0, 1)$ as well as the χ^2 distribution $\chi^2(5)$ are considered to generate noise features. The details of these synthetic datasets are summarized in Table A.4. The reported results in Table A.5 are the averaged CERs of each algorithm. We also report the F1score values of feature selection of each algorithm in Table A.6. It can be seen that on all synthetic datasets with non-linear cluster structures, the proposed method (SWKKM) outperforms other peer algorithms in clustering according to its lowest CERs. Moreover, it also successfully selects all relevant features according to the highest F1scores, while the peer algorithms fail.

At last, we discuss more about *data9*. It consists of 200 sample points ($n = 200$) and 120 features ($p = 120$). The sample is drawn from 10 different clusters ($k = 10$) and each of them has 20 sample points. The first 20 features are relevant to the ground truth structure ($d = 20$), while the remaining 100 features follow the standard normal distribution. The ground truth structure is generated in the following way. We draw 10 true cluster centers $\boldsymbol{c}_l$ uniformly along the surface of a unit sphere $S_{19} = \{x \mid \|x\|_2 = 1\} \subset \mathbb{R}^{20}$, and assign ground truth labels Z_i to each observation X_i uniformly. A point assigned to cluster l is drawn from the von-Mises-Fisher distribution normalized to lie on S_{19} with mean direction $\boldsymbol{c}_l$ and $\kappa = 30$. That is,

$$\boldsymbol{c}_1, \cdots, \boldsymbol{c}_k \sim U(S_{19}), \quad Z_i \sim U\{1, \cdots, k\}, \quad X_i \mid Z_i \sim VMF(\boldsymbol{c}_{Z_i}, \kappa).$$

Finally combined with the noise part, the *data9* is generated. We run each algorithm 10 times and report the average value of CERs in Table A.5. We

Table A.4: Details of synthetic datasets used for SWKKM.

Dataset	Ground truth	Noise	k	n	p	d
data1	Two moons	$\mathcal{N}(0,1)$	2	200	12	2
data2	Cassini	$\mathcal{N}(0,1)$	3	300	12	2
data3	Atom	$\mathcal{N}(0,1)$	2	200	13	3
data4	Chainlink	$\mathcal{N}(0,1)$	2	200	13	3
data5	Two moons	$\chi^2(5)$	2	200	12	2
data6	Cassini	$\chi^2(5)$	3	300	12	2
data7	Atom	$\chi^2(5)$	2	200	13	3
data8	Chainlink	$\chi^2(5)$	2	200	13	3
data9	VMF	$\mathcal{N}(0,1)$	10	200	120	20
data10	VMF	$\chi^2(5)$	10	200	120	20

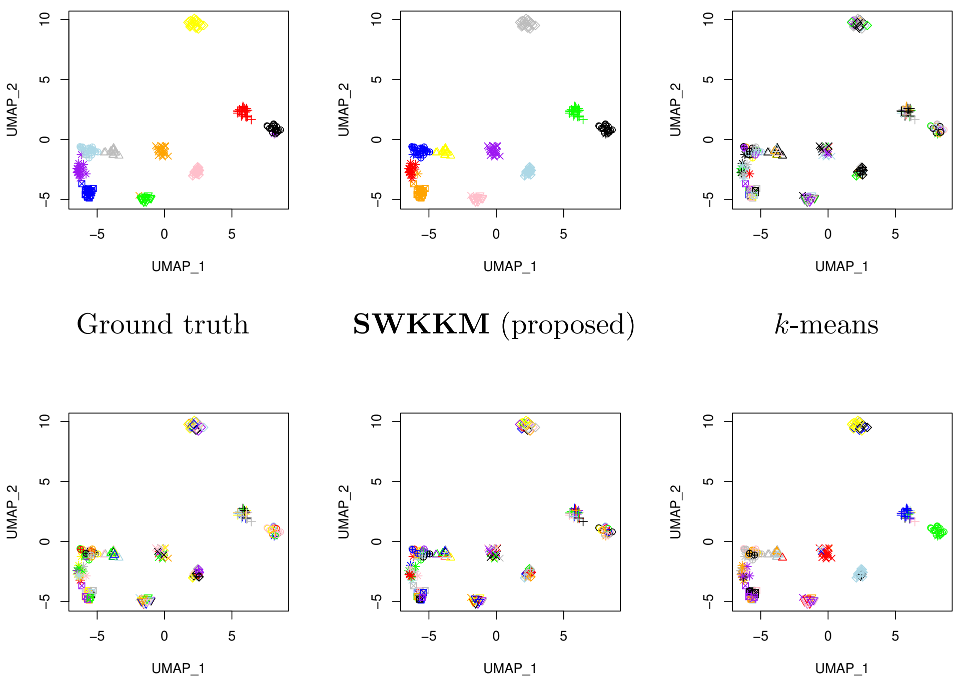
Table A.5: Averaged CERs of different methods on synthetic datasets.

Dataset	k-means	Weighted kernel k-means	IF-PCA	Sparse k-means	**SWKKM** (proposed)
data1	0.279	0.500	0.095	0.333	**0.058**
data2	0.413	0.444	0.237	0.335	**0.149**
data3	0.499	0.499	0.375	0.499	**0.274**
data4	0.493	0.501	0.475	0.502	**0.412**
data5	0.502	0.500	0.500	0.498	**0.082**
data6	0.451	0.443	0.325	0.463	**0.159**
data7	0.500	0.500	0.460	0.497	**0.267**
data8	0.500	0.499	0.490	0.502	**0.342**
data9	0.243	0.178	0.107	0.179	**0.008**
data10	0.259	0.177	0.103	0.183	**0.009**

Table A.6: Averaged Precision, Recall and F1score indexes of different methods on synthetic datasets.

Dataset	Precision			Recall			F1score		
	IF-PCA	SKM	SWKKM	IF-PCA	SKM	SWKKM	IF-PCA	SKM	SWKKM
data1	0.278	0.167	1.000	0.590	1.000	1.000	0.368	0.286	**1.000**
data2	0.000	0.167	1.000	0.000	1.000	1.000	0.000	0.286	**1.000**
data3	0.367	0.231	1.000	1.000	1.000	1.000	0.530	0.375	**1.000**
data4	0.333	0.231	1.000	0.500	1.000	1.000	0.400	0.375	**1.000**
data5	0.750	0.167	1.000	1.000	1.000	1.000	0.857	0.286	**1.000**
data6	0.429	0.167	1.000	1.000	1.000	1.000	0.600	0.286	**1.000**
data7	0.000	0.231	1.000	0.000	1.000	1.000	0.000	0.375	**1.000**
data8	0.000	0.231	1.000	0.000	1.000	1.000	0.000	0.375	**1.000**
data9	0.144	0.167	1.000	0.267	1.000	1.000	0.187	0.286	**1.000**
data10	0.333	0.167	1.000	0.667	1.000	1.000	0.444	0.286	**1.000**

also illustrate the clustering result of each algorithm using a visualization technique (UMAP, (McInnes et al. 2018)) in Figure A.3. It can be seen that the clustering result of the proposed method is most similar to the ground truth, while other methods almost fail to obtain the true cluster structure.



Ground truth

SWKKM (proposed)

k-means

Sparse *k*-means

Weighted kernel *k*-means

IF-PCA

Figure A.3: The clustering results for *data9* of different methods. The x-axis and y-axis are the first two dimensions of UMAP. The color of points represents which cluster the point belongs to.

Appendix B

Appendix for Chapter 3

B.1 Details of Algorithm 3.2

In this section, we provide technical details of Algorithm 3.2 proposed in Chapter 3[1].

B.1.1 Derivation of updating $\mathrm{M}^{(r+1)}$ for $J = J_0$

For $J = J_0$, given $\mathrm{U}^{(r+1)}$, the update $\mathrm{M}^{(r+1)}$ is given by the solution of

$$\min_{\mathrm{M}} \|\widehat{\mathrm{X}}_{(j)} - \mathrm{U}^{(r+1)}\mathrm{M}\|_F^2 + \lambda \sum_{j=1}^{p} \mathbb{1}(\|\mathrm{M}_{(j)}\| > 0).$$

Because $\|\widehat{\mathrm{X}}_{(j)} - \mathrm{U}^{(r+1)}\mathrm{M}\|_F^2 = \sum_{j=1}^{p} \|\widehat{\mathrm{X}}_{(j)} - \mathrm{U}^{(r+1)}\mathrm{M}_{(j)}\|^2$, we can separately solve the minimization problem in each feature, that is, for any $j = 1, \ldots, p$,

$$\min_{\mathrm{M}_{(j)}} \|\widehat{\mathrm{X}}_{(j)} - \mathrm{U}^{(r+1)}\mathrm{M}_{(j)}\|^2 + \lambda \mathbb{1}(\|\mathrm{M}_{(j)}\| > 0).$$

If the solution $\widehat{\mathrm{M}}_{(j)} \neq 0$, then $\mathbb{1}(\|\mathrm{M}_{(j)}\| > 0) = 1$ and the KKT condition implies that

$$\widehat{\mathrm{M}}_{(j)} = (\mathrm{U}^{(r+1),T}\mathrm{U}^{(r+1)})^{-1}\mathrm{U}^{(r+1),T}\widehat{\mathrm{X}}_{(j)}.$$

If the solution $\widehat{\mathrm{M}}_{(j)} = 0$, then the corresponding value of objective function is $\|\widehat{\mathrm{X}}_{(j)}\|^2$, which should be smaller than the objective function at any non-zero point. Therefore, there must be

$$\|\widehat{\mathrm{X}}_{(j)} - \mathrm{U}^{(r+1)}\mathrm{V}_{(j)}\|^2 + \lambda \geq \|\widehat{\mathrm{X}}_{(j)}\|^2,$$

where $\mathrm{V}_{(j)} = (\mathrm{U}^{(r+1),T}\mathrm{U}^{(r+1)})^{-1}\mathrm{U}^{(r+1),T}\widehat{\mathrm{X}}_{(j)}$.

[1] Throughout Appendix B, we use $\|\cdot\|$ to express the l_2 norm $\|\cdot\|_2$.

B.1.2 Derivation of updating $\mathrm{M}^{(r+1)}$ for $J = J_1$

For $J = J_1$, given $\mathrm{U}^{(r+1)}$, the update $\mathrm{M}^{(r+1)}$ is the solution of

$$\min_{\mathrm{M}} \|\widehat{\mathrm{X}} - \mathrm{U}^{(r+1)}\mathrm{M}\|_F^2 + \lambda \sum_{j=1}^{p} w_j \|\mathrm{M}_{(j)}\|,$$

where the objective function is denoted by $f(\mathrm{M})$ in Eq. (3.6). Since it is not easy to derive an explicit solution, we instead apply the MM algorithm again to obtain $\mathrm{M}^{(r+1)}$. At any point $\mathrm{M}^{(r_s)}$ ($s \in \mathbb{N}$), the function h in Eq. (3.7) is given by

$$h(\mathrm{M} \mid \mathrm{M}^{(r_s)}) = \|\widehat{\mathrm{X}} - \mathrm{U}^{(r+1)}\mathrm{M}\|_F^2 + \lambda \sum_{j=1}^{p} w_j \left(\frac{\|\mathrm{M}_{(j)}\|^2}{2\|\mathrm{M}_{(j)}^{(r_s)}\|} + \frac{1}{2}\|\mathrm{M}_{(j)}^{(r_s)}\| \right).$$

Based on the basic equality, we have for each $j = 1, \ldots, p$,

$$\frac{\|\mathrm{M}_{(j)}\|^2}{2\|\mathrm{M}_{(j)}^{(r_s)}\|} + \frac{1}{2}\|\mathrm{M}_{(j)}^{(r_s)}\| \geq \|\mathrm{M}_{(j)}\|,$$

where the equality holds if and only if $\mathrm{M}_{(j)}^{(r_s)} = \mathrm{M}_{(j)}$. It follows that

$$h(\mathrm{M} \mid \mathrm{M}^{(r_s)}) \geq f(\mathrm{M}) \text{ and } h(\mathrm{M}^{(r_s)} \mid \mathrm{M}^{(r_s)}) = f(\mathrm{M}^{(r_s)}),$$

which means that the domination condition and tangency condition are satisfied and $h(\mathrm{M} \mid \mathrm{M}^{(r_s)})$ majorizes $f(\mathrm{M})$ at any $\mathrm{M}^{(r_s)}$. Now we can apply the MM algorithm in the following way. Starting from $\mathrm{M}^{(r_0)}$, the $(s+1)$-th iteration includes: (i) construct the majorization function $h(\mathrm{M} \mid \mathrm{M}^{(r_s)})$ with current $\mathrm{M}^{(r_s)}$; (ii) update $\mathrm{M}^{(r_{s+1})}$ by minimizing $h(\mathrm{M} \mid \mathrm{M}^{(r_s)})$, the solution of which can be easily derived by KKT condition in each feature, that is, for any $j = 1, \ldots, p$,

$$\mathrm{M}_{(j)}^{(r_{s+1})} = \left(\mathrm{U}^{(r+1),T}\mathrm{U}^{(r+1)} + \frac{\lambda w_j}{2\|\mathrm{M}_{(j)}^{(r_s)}\|} \cdot \mathrm{I}_k \right)^{-1} \mathrm{U}^{(r+1),T}\widehat{\mathrm{X}}_{(j)}.$$

This procedure ensures that $f(\mathrm{M}^{(r_{s+1})}) \leq f(\mathrm{M}^{(r_s)})$ for any $s \in \mathbb{N}$. As discussed in Remark 3.1, there is no need to exactly minimize $f(\mathrm{M})$. Instead, reducing $f(\mathrm{M})$ is enough. Therefore, to simplify the computation, we only conduct once iteration about s, that is, we start from $\mathrm{M}^{(r_0)} = \mathrm{M}^{(r)}$ and update the j-th column of $\mathrm{M}^{(r+1)}$ by

$$\left(\mathrm{U}^{(r+1),T}\mathrm{U}^{(r+1)} + \frac{\lambda w_j}{2\|\mathrm{M}_{(j)}^{(r)}\|} \cdot \mathrm{I}_k \right)^{-1} \mathrm{U}^{(r+1),T}\widehat{\mathrm{X}}_{(j)}.$$

B.1.3 Comparison with other method for $J = J_1$

As explained in Remark 3.2, updating $\mathrm{M}^{(r+1)}$ for $J = J_1$ is equivalent to the group lasso regression. Specifically, minimizing $f(\mathrm{M})$ is equivalent to minimizing $f_j(\mathrm{M}_{(j)})$ for each $j = 1, \ldots, p$, where

$$f_j(\mathrm{M}_{(j)}) = \|\widehat{\mathrm{X}}_{(j)} - \mathrm{U}^{(r+1)}\mathrm{M}_{(j)}\|_2^2 + \lambda w_j \|\mathrm{M}_{(j)}\|.$$

It can be viewed as a regression model of response $\widehat{\mathrm{X}}_{(j)}$ on design matrix $\mathrm{U}^{(r+1)}$ with a group lasso penalty $\|\mathrm{M}_{(j)}\|$, where the number of groups is one. For simplification of notations, we write y for $\widehat{\mathrm{X}}_{(j)}$, write U for $\mathrm{U}^{(r+1)}$ and write β for $\mathrm{M}_{(j)}$.

Following the method of Yang & Zou (2015), we can construct a majorization function $\tilde{h}_j(\beta \mid \beta^{(s)})$ for $f_j(\beta)$ at any point $\beta^{(s)}$ via quadratic approximation of the first term of $f_j(\beta)$. Denote $l(\beta) = \|y - \mathrm{U}\beta\|^2$. Because

$$l(\beta) \leq l(\beta^{(s)}) + (\beta - \beta^{(s)})^T \nabla l(\beta^{(s)}) + \frac{1}{2}(\beta - \beta^{(s)})^T \mathrm{H}(\beta - \beta^{(s)}),$$

where $\nabla l(\beta^{(s)}) = -2\mathrm{U}^T(y - \mathrm{U}\beta^{(s)})$ and $\mathrm{H} = -2\mathrm{U}^T\mathrm{U}$, we can define

$$\tilde{h}_j(\beta \mid \beta^{(s)}) = l(\beta^{(s)}) + (\beta - \beta^{(s)})^T \cdot \left(-2\mathrm{U}^T\right) \cdot (y - \mathrm{U}\beta^{(s)}) + \frac{\gamma}{2}\|\beta - \beta^{(s)}\|^2 + \lambda w_j \|\beta\|,$$

where $\gamma = 2\max_l \sum_{i=1}^n u_{il}$ is the largest size of clusters associated with U. The minimizer of $\tilde{h}_j(\beta \mid \beta^{(s)})$ is thus give by

$$\beta^{(s+1)} = \tilde{\beta} \cdot \left(1 - \frac{\lambda w_j/\gamma}{\|\tilde{\beta}\|}\right)_+,$$

where $\tilde{\beta} = \beta^{(s)} + \frac{2}{\gamma}\mathrm{U}^T(y - \mathrm{U}\beta^{(s)})$ is the gradient descent update of $l(\beta)$ and $(\cdot)_+ = \max(\cdot, 0)$. Therefore, we propose Algorithm B.1 for $J = J_1$ based on the quadratic approximation.

Next, we compare Algorithm 3.2 and Algorithm B.1 via numerical experiments on synthetic complete datasets. Figure B.1 illustrates regularization paths of these two algorithms on datasets with $p = 10$ and $p = 100$, and Figure B.2 shows the convergence and computational time in the case of $p = 100$. It can be seen that the paths of two algorithms are almost the same, while Algorithm B.1 needs fewer iterations and thus less computational time.

Algorithm B.1 Regularized k-means clustering using quadratic approximation

```
Input: complete data matrix $\widehat{\mathrm{X}}$, number of clusters $k$.
Parameters: regularized parameter $\lambda$, weights $\{w_j\}$
  Initialize $\mathrm{M}^{(0)}$
  while Loss function (3.5) does not converge do
      a: Given $\mathrm{M}^{(r)}$, update $\mathrm{U}^{(r+1)}$ by: for any $i = 1, \ldots, n$
          $u_{il^*}^{(r+1)} = \begin{cases} 1 & \text{if } l^* = \arg\min_{1 \le l \le k} \|\widehat{X}_i - \mathrm{M}_l^{(r)}\|^2 \\ 0 & \text{else} \end{cases}$
      b: Given $\mathrm{U}^{(r+1)}$, update $\mathrm{M}^{(r+1)}$ by: for any $j = 1, \ldots, p$
          $\mathrm{M}_{(j)}^{(r+1)} = \tilde{\mathrm{V}}_{(j)} \cdot \left(1 - \frac{\lambda w_j / \gamma}{\|\tilde{\mathrm{V}}_{(j)}\|}\right)_+ ,$
          where $\tilde{\mathrm{V}}_{(j)} = \mathrm{M}_{(j)}^{(r)} + \frac{2}{\gamma} \mathrm{U}^{(r+1),T} \cdot \left(\widehat{\mathrm{X}}_{(j)} - \mathrm{U}^{(r+1)} \mathrm{M}_{(j)}^{(r)}\right)$
          $\gamma = 2 \cdot \max\left[\mathrm{diag}\left(\mathrm{U}^{(r+1),T} \mathrm{U}^{(r+1)}\right)\right]$
  end while
Output: $\mathrm{U}^{(r+1)}$ and $\mathrm{M}^{(r+1)}$
```

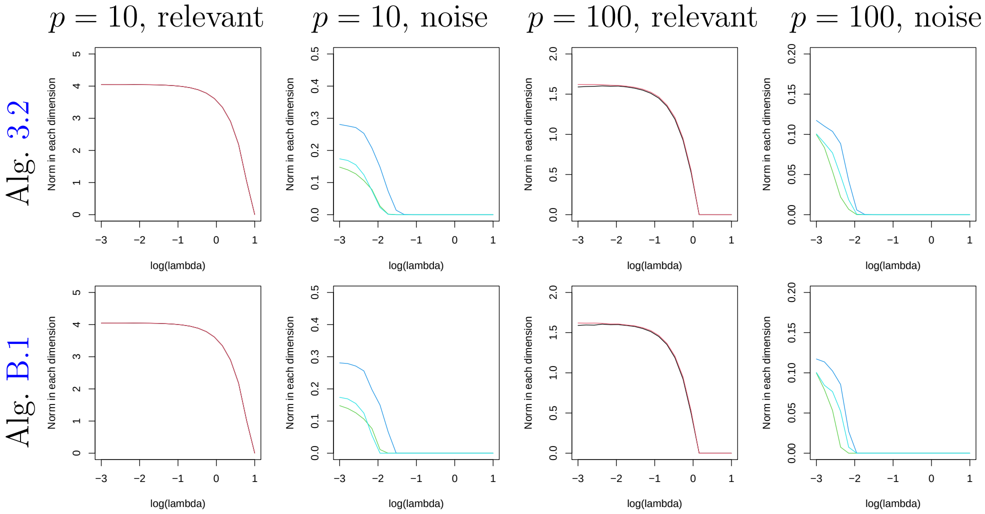
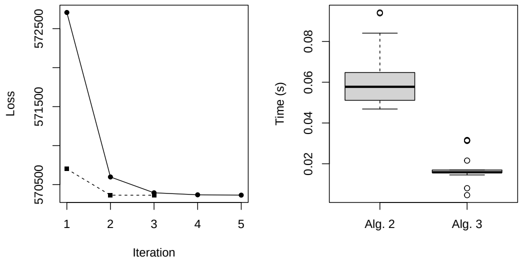


Figure B.1: Regularization paths of Algorithm 3.2 (top) and Algorithm B.1 (bottom). The x-axis is the $\log(\lambda)$ and the y-axis is $\|\mathrm{M}_{(j)}\|$. The four columns are for two relevant features in case of $p = 10$, three noise features in case of $p = 10$, two relevant features in case of $p = 100$ and three noise features in case of $p = 100$.



(a) (b)

Figure B.2: (a) Convergence of Algorithm 3.2 (solid) and Algorithm B.1 (dotted) in the case of $p = 100$. (b) Comparison of computational time, denoted by *Alg.2* and *Alg.3* for Algorithm 3.2 and Algorithm B.1, respectively.

B.2 Derivation of BIC

In this section, we provide technical details of deriving the expression of BIC given in Eq. (3.10). We first consider the classification likelihood (Fraley & Raftery 2002) to formulate the k-means likelihood. Let $\{X_i\}_{i=1}^n$ be i.i.d. sample, $\mathrm{U} = (u_{il})_{n\times k} \in \{0,1\}^{n\times k}$ be the indicators of membership of $\{X_i\}_{i=1}^n$ and $\mathrm{U}\mathbf{1}_k = \mathbf{1}_n$. Denote by $\phi_p(\cdot \mid \mathrm{M}_l)$ the density function of Gaussian distribution in $\mathbb{R}^p$ with mean vector $\mathrm{M}_l = (\mu_{l1}, \dots, \mu_{lp})$ and covariance matrix $\sigma^2\mathrm{I}_p$, where σ^2 is fixed. Write $\mathrm{M} = (\mu_{lj})_{k\times p}$. The classification likelihood of X_i is given by

$$\begin{aligned} l(X_i \mid \mathrm{U}, \mathrm{M}) &= \prod_{l=1}^k [\phi_p(X_i \mid \mathrm{M}_l)]^{u_{il}} \\ &= \prod_{l=1}^k \left[(2\pi\sigma^2)^{-\frac{p}{2}} \exp\left(-\frac{\sum_{j=1}^p (x_{ij} - \mu_{lj})^2}{2\sigma^2}\right)\right]^{u_{il}} \end{aligned}$$

Now we consider the missing data. Assume that for $X_i = (x_{i1}, \dots, x_{ip})$, any element x_{ij} is missing completely at random (MCAR) and X_i would be partially observed. As in Section 3.4, we use a binary random variable r_{ij} to indicate whether x_{ij} is observed. That is, $r_{ij} = 1$ if x_{ij} is observed, 0 otherwise. Write $\mathrm{R}_i = (r_{i1}, \dots, r_{ip}) \in \{0,1\}^p$. The MCAR mechanism means that R_i is independent with X_i. Because the covariance matrix is $\sigma^2\mathrm{I}_p$, we have $\phi_p(X_i \mid \mathrm{M}_l) = \prod_{j=1}^p \phi(x_{ij} \mid \mu_{lj})$, where $\phi(\cdot \mid \mu_{lj})$ is the density function of Gaussian distribution in $\mathbb{R}$ with mean μ_{lj} and variance σ^2. Then the likelihood of X_i can be written as

$$\begin{aligned} l(X_i \mid \mathrm{U}, \mathrm{M}) &= \prod_{l=1}^k \left[\prod_{j=1}^p \phi(x_{ij} \mid \mu_{lj})\right]^{u_{il}} \\ &= \prod_{l=1}^k \left[\prod_{j:r_{ij}=1} \phi(x_{ij} \mid \mu_{lj}) \cdot \prod_{j:r_{ij}=0} \phi(x_{ij} \mid \mu_{lj})\right]^{u_{il}} \\ &= \prod_{l=1}^k \left[\prod_{j:r_{ij}=1} \phi(x_{ij} \mid \mu_{lj})\right]^{u_{il}} \cdot \prod_{l=1}^k \left[\prod_{j:r_{ij}=0} \phi(x_{ij} \mid \mu_{lj})\right]^{u_{il}} . \end{aligned}$$

The likelihood of partially observed part, denoted by X_i^{obs}, is thus equivalent to the density of marginal distribution of $\{x_{ij} \mid r_{ij} = 1\}$, which is given by

$$l(X_i^{obs} \mid \mathrm{U}, \mathrm{M}, \mathrm{R}_i) = \prod_{l=1}^k \left[\prod_{j:r_{ij}=1} \phi(x_{ij} \mid \mu_{lj})\right]^{u_{il}} .$$

Therefore, the likelihood of partially observed sample $\{X_1^{obs}, \dots, X_n^{obs}\}$ is given by

$$l_n(X_1^{obs}, \dots, X_n^{obs} \mid \mathrm{R}, \mathrm{U}, \mathrm{M}) = \prod_{i=1}^{n} \prod_{l=1}^{k} \left[(2\pi\sigma^2)^{-\frac{\|\mathrm{R}_i\|}{2}} \exp\left(-\frac{\sum_{j=1}^{p} r_{ij}(x_{ij} - \mu_{lj})^2}{2\sigma^2} \right) \right]^{u_{il}}.$$

Then we have

$$\mathrm{BIC} = \sum_{i=1}^{n} \|\mathrm{R}_i\| \log(2\pi\sigma^2) + \frac{1}{\sigma^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{l=1}^{k} r_{ij} u_{il} (x_{ij} - \mu_{lj})^2 + \log(n) \cdot df.$$

The first term is a fixed constant and the second term is equivalent to $\|\mathcal{P}_\Omega(\mathrm{X} - \mathrm{UM})\|_F^2$, then we can write BIC to be

$$\mathrm{BIC} = \|\mathcal{P}_\Omega(\mathrm{X} - \mathrm{UM})\|_F^2 + \log(n) \cdot df,$$

where df is the number of independent parameters, which is simply kd with $d = \sum_{j=1}^{p} \mathbb{1}(\|\mathrm{M}_{(j)}\| > 0)$. Note that df can be further approximated by using the effective degree of freedom as discussed in Hofmeyr (2020).

B.3 Supplementary for numerical experiments

In this section, we provide more details and results of numerical experiments for Section 3.5.

B.3.1 Settings of missingness mechanisms

Through the numerical experiments, we consider four types of procedures for generating missingness. For MAR and MNAR1 mechanisms, different parameters used to meet the total proportion of missingness are summarized in Table B.1.

B.3.2 Comparison of random and non-random initialization

For l_0 type of proposed method, since it is sensitive to the initialization, we here compare the random initialization and non-random initialization. Specifically, we consider the *sparse initialization*, which is also used in Raymaekers & Zamar (2022). First, based on the estimated cluster centers of k-POD clustering, we rank all p features in a decreasing order by the l_2 norms of k-POD estimator in each feature. Then by retaining only the leading 1%,

Table B.1: Different parameters to meet total missing proportion

Dataset	Missing Proportion	MAR		MNAR1	
		ψ_1	ψ_2	ϕ_1	ϕ_2
$p = 10$	10%	1.80	3.0	1.5	3.0
	20%	0.55	3.0	0.6	3.0
	30%	0.25	3.0	0.3	3.0
$p = 100$, $a = 0.8$	10%	2.0	2.0	2.5	2.0
	20%	0.8	2.0	0.9	2.0
	30%	0.4	2.0	0.45	2.0
$p = 100$, $a = 1$	10%	2.5	2.0	2.5	2.0
	20%	0.9	2.0	0.9	2.0
	30%	0.45	2.0	0.45	2.0

2%, 5%, 10%, 15%, 20%, 30%, 40%, 50%, 100% features, we can get 10 sparse versions of k-POD estimator. These 10 sparse estimators would serve as 10 initialization points for the proposed method. For the random initialization, we use 100 initialization points.

Table B.2 illustrates the comparison results between random initialization and sparse initialization. For the random initialization, since we consider two strategies, only the best results are reported. For the sparse initialization, when $p = 10$, we use the sequence $\{10\%, 20\%, \ldots, 100\%\}$ to generate the 10 sparse initialization points. Moreover, for the dataset with $p = 100$, the setting of $d = 10$ and $a = 0.8$ is used.

It can be seen that the sparse initialization generally provides comparable results, especially in the case of $p = 100$, while it only uses 10 initial points and needs less computational time. Therefore, the sparse initialization can be used as a faster substitute for random initialization when the number of features is large, as it requires fewer initialization points.

B.3.3 Sensitivity analysis of tuning parameter

We analyze the sensitivity of the regularization parameter based on the case of $p = 100$. Figure B.3 illustrates the results of instability and BIC under MCAR mechanism with missing proportion 30%, where the reported values are the average of 10 repetitions. It can be seen that a suitable λ can reduce the value of MSE and provide a reasonable set of features that contribute to clustering. Moreover, the instability is more sensitive to λ than BIC.

We further report the comparison of the instability and BIC criteria for selecting λ in the case of $p = 10$ in Table B.3. It can be seen that the instability is stable in various settings, which is similar to the case of $p = 100$, while BIC almost fails. The main reason is that there are only two relevant

Table B.2: Comparison of random initialization and sparse initialization for l_0 type of proposed method

Dataset	Missing mechanism	Missing proportion	MSE		CER	
			random	sparse	random	sparse
$p = 10$	MCAR	10%	**0.025 (0.01)**	0.110 (0.03)	**0.123 (0.01)**	0.124 (0.01)
		20%	**0.079 (0.03)**	0.296 (0.07)	**0.186 (0.01)**	0.190 (0.00)
		30%	**0.097 (0.00)**	0.557 (0.10)	**0.241 (0.01)**	0.242 (0.01)
		40%	**1.139 (2.46)**	2.406 (5.38)	**0.285 (0.01)**	0.297 (0.03)
		50%	22.601 (6.93)	**4.466 (7.09)**	**0.345 (0.01)**	0.353 (0.04)
$p = 100$	MCAR	10%	0.134 (0.02)	**0.131 (0.02)**	0.089 (0.01)	**0.086 (0.00)**
		20%	0.153 (0.03)	**0.149 (0.03)**	0.113 (0.00)	**0.108 (0.01)**
		30%	7.948 (5.29)	**2.285 (3.65)**	0.245 (0.04)	**0.177 (0.04)**
		40%	26.469 (5.00)	**18.428 (8.16)**	0.375 (0.03)	**0.303 (0.05)**
		50%	36.284 (2.77)	**26.843 (4.60)**	0.376 (0.01)	**0.329 (0.01)**
Usoskin	MNAR	73%	-	-	0.167 (0.03)	**0.133 (0.04)**

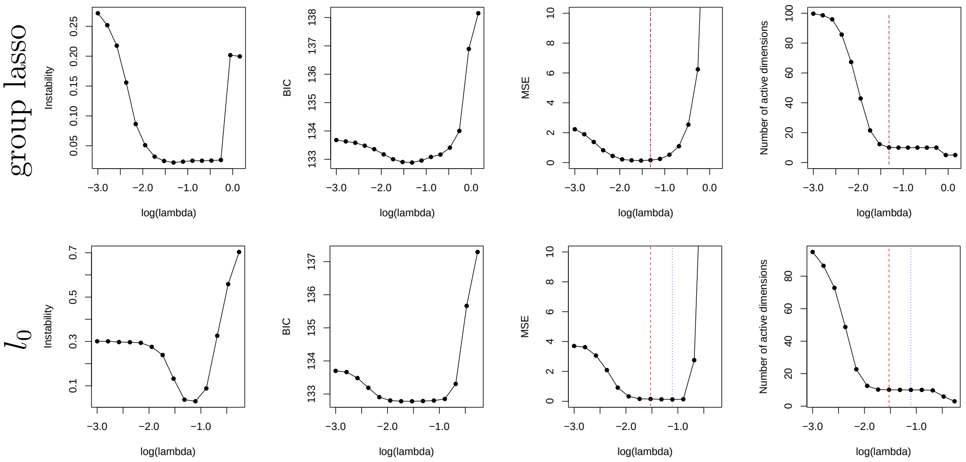


Figure B.3: Comparison of instability and BIC criteria for selecting λ. The top and bottom rows are for group lasso and l_0 types of proposed method, respectively. The red dashed lines denote the choice of BIC, while the blue dotted lines denote instability.

features in this case, the decrease of active features has more influence on increasing the loss than decreasing the degree of freedom.

Table B.3: MSE (number of active features in brackets) of proposed method using different criteria for selecting λ ($p = 10$)

Missing mechanism	Missing proportion	group lasso		l_0	
		Instability	BIC	Instability	BIC
MCAR	10%	0.118 (3)	1.508 (10)	0.038 (2)	1.324 (7)
	20%	0.872 (6)	2.767 (10)	0.079 (2)	4.677 (9)
	30%	1.853 (7)	8.467 (10)	0.097 (2)	16.547 (9)
	40%	3.160 (7)	26.199 (10)	1.139 (2)	24.100 (8)
	50%	4.732 (3)	30.416 (10)	22.601 (4)	31.611 (9)
MAR	10%	0.364 (3)	1.764 (10)	0.203 (2)	1.335 (5)
	20%	0.298 (2)	5.501 (10)	0.117 (2)	5.022 (8)
	30%	0.484 (2)	2.861 (8)	0.115 (2)	4.487 (5)
MNAR1	10%	1.151 (5)	5.100 (10)	0.462 (2)	5.576 (10)
	20%	3.932 (2)	12.476 (10)	0.283 (2)	15.486 (10)
	30%	2.301 (4)	21.715 (10)	0.210 (2)	21.032 (10)
MNAR2	10%	2.006 (3)	6.322 (10)	0.691 (2)	6.384 (10)
	20%	4.901 (10)	21.431 (10)	2.346 (2)	21.598 (10)
	30%	24.829 (3)	45.131 (10)	9.733 (2)	47.213 (10)

B.3.4 More results of comparable experiments

Finally, we report the result of comparing the performance of the proposed method with other methods in the case $p = 100$ and $a = 1$, which is an easier clustering task with more separable cluster centers. Table B.4, Table B.5 and Table B.6 illustrate the comparison of MSE, CER and predictive CER, respectively.

Table B.4: MSE (standard deviations in brackets) of different methods ($p = 100$ and $a = 1$)

Missing mechanism	Missing proportion	Mice	k-POD	Reg. k-POD (group lasso)	Reg. k-POD (l_0)
MCAR	10%	1.286 (0.09)	1.430 (0.09)	0.126 (0.02)	**0.109 (0.02)**
	20%	1.462 (0.10)	1.870 (0.14)	0.206 (0.04)	**0.156 (0.03)**
	30%	1.788 (0.11)	3.063 (0.49)	0.407 (0.10)	**0.280 (0.08)**
	40%	2.272 (0.14)	19.121 (2.43)	**1.918 (0.30)**	2.675 (1.60)
	50%	**3.267 (0.23)**	36.512 (3.54)	5.546 (2.91)	25.073 (4.03)
MAR	10%	1.338 (0.13)	1.516 (0.14)	0.150 (0.04)	**0.131 (0.03)**
	20%	1.517 (0.11)	1.842 (0.16)	0.140 (0.03)	**0.126 (0.02)**
	30%	1.771 (0.14)	3.117 (0.73)	0.204 (0.05)	**0.164 (0.03)**
MNAR1	10%	25.983 (0.58)	26.039 (0.52)	3.073 (0.16)	**1.873 (0.13)**
	20%	32.579 (0.70)	33.187 (0.73)	3.109 (0.17)	**1.738 (0.33)**
	30%	25.673 (0.56)	27.698 (0.83)	2.139 (0.20)	**1.324 (0.37)**
MNAR2	10%	31.768 (0.62)	31.161 (0.61)	4.696 (0.18)	**2.693 (0.22)**
	20%	101.579 (0.97)	99.327 (1.280)	**40.286 (0.04)**	99.507 (1.31)

Table B.5: CER (standard deviations in brackets) of different methods ($p = 100$ and $a = 1$)

Missing mechanism	Missing proportion	Mice	k-POD	Reg. k-POD (group lasso)	Reg. k-POD (l_0)
MCAR	10%	**0.044 (0.00)**	0.046 (0.00)	0.050 (0.00)	0.049 (0.00)
	20%	**0.064 (0.00)**	0.072 (0.00)	0.091 (0.00)	0.092 (0.00)
	30%	**0.092 (0.00)**	0.124 (0.02)	0.147 (0.00)	0.147 (0.00)
	40%	**0.126 (0.00)**	0.287 (0.02)	0.186 (0.00)	0.236 (0.02)
	50%	**0.170 (0.01)**	0.364 (0.01)	0.259 (0.01)	0.356 (0.02)
MAR	10%	0.052 (0.00)	0.056 (0.01)	**0.047 (0.00)**	0.051 (0.00)
	20%	0.063 (0.00)	0.074 (0.01)	**0.063 (0.00)**	0.064 (0.01)
	30%	0.086 (0.01)	0.127 (0.02)	0.088 (0.00)	**0.086 (0.01)**
MNAR1	10%	0.063 (0.00)	0.058 (0.00)	**0.053 (0.00)**	0.056 (0.00)
	20%	0.079 (0.00)	0.082 (0.01)	**0.079 (0.01)**	0.091 (0.01)
	30%	**0.102 (0.00)**	0.150 (0.02)	0.139 (0.01)	0.152 (0.01)
MNAR1	10%	0.064 (0.00)	0.056 (0.00)	**0.051 (0.01)**	0.056 (0.00)
	20%	0.124 (0.00)	**0.117 (0.01)**	0.746 (0.00)	0.149 (0.01)

Table B.6: Predictive CER (standard deviations in brackets) of different methods ($p = 100$ and $a = 1$)

Missing mechanism	Missing proportion	Mice	k-POD	Reg. k-POD (group lasso)	Reg. k-POD (l_0)
MCAR	10%	0.030 (0.01)	0.033 (0.01)	**0.027 (0.01)**	0.028 (0.01)
	20%	0.030 (0.01)	0.035 (0.01)	**0.026 (0.01)**	0.027 (0.01)
	30%	0.032 (0.01)	0.043 (0.01)	0.029 (0.01)	**0.028 (0.01)**
	40%	0.032 (0.01)	0.190 (0.02)	**0.027 (0.01)**	0.043 (0.02)
	50%	**0.036 (0.01)**	0.280 (0.02)	0.043 (0.03)	0.234 (0.04)
MAR	10%	0.030 (0.01)	0.031 (0.01)	0.024 (0.01)	**0.023 (0.01)**
	20%	0.028 (0.01)	0.036 (0.01)	0.028 (0.01)	**0.028 (0.01)**
	30%	0.032 (0.01)	0.042 (0.01)	0.025 (0.01)	**0.024 (0.01)**
MNAR1	10%	0.039 (0.01)	0.042 (0.01)	**0.028 (0.01)**	0.029 (0.01)
	20%	0.040 (0.01)	0.045 (0.01)	**0.034 (0.01)**	0.035 (0.01)
	30%	0.036 (0.01)	0.052 (0.01)	**0.034 (0.01)**	0.036 (0.01)
MNAR2	10%	0.046 (0.01)	0.046 (0.01)	**0.033 (0.01)**	0.037 (0.01)
	20%	**0.091 (0.01)**	0.096 (0.02)	0.303 (0.02)	0.099 (0.02)

List of publications

Guan, X. & Terada, Y. (2023), ‘Sparse kernel k-means for high-dimensional data’, *Pattern Recognition* **144**, 109873.

Guan, X. & Terada, Y. (2024), ‘Regularized k-POD clustering for high-dimensional missing data’, *Submitted to Statistics and Computing.*

Acknowledgments

This thesis owes its existence to the invaluable contributions of my esteemed supervisor, Assoc. Prof. Dr. Yoshikazu Terada. On my Ph.D. journey, nothing is better than having his professional, patient, dedicated, detailed, and all-encompassing guidance. Neither English, Japanese nor Chinese can express my gratitude to him. I can only strive to live up to his years of training through relentless effort in my future research career.

I am deeply grateful to committee members, including Prof. Joe Suzuki, Prof. Masayuki Uchida and Prof. Tomoyuki Sugimoto, for carefully reviewing my work. Their insightful and constructive suggestions and comments have significantly improved the quality of this thesis.

My gratitude also goes to Prof. Yutaka Kano and Prof. Hiroshi Yadohisa, who helped me apply for scholarships and accepted me as a research assistant in their labs. This support enabled me to devote myself fully to research without financial concerns.

I also thank Dr. Bingyuan Zhang, who provided crucial assistance and meaningful discussions for proposing the methodology and algorithm in Chapter 2 and reviewing the manuscript for submission, and thank Asst. Prof. Kosuke Morikawa and Kenji Beppu for their generous feedback on the work of Chapter 3, which greatly enhanced the submitted manuscript.

Moreover, I would like to thank members of Kano Lab, Joe Lab and Yadohisa Lab, who inspired me to think of my research and other statistical problems from different perspectives. Thanks are also extended to all faculty and staff members in the Graduate School of Engineering Science for their professional and warm-hearted service.

Beyond Osaka University, I would like to express my special thanks to Assoc. Prof. Yujie Jiang, Dr. Maoxin Zhang and Ms. Xuelei Zhang. Their companionship, understanding and encouragement supported me through numerous tough, painful and desperate moments. Additionally, the kindness I received from anonymous reviewers encouraged me to keep going.

This thesis is supported by China Scholarship Council (NO. 202108050077) and, most importantly, by the love of my parents.

Bibliography

Almeida, J. S. & Prieto, C. A. (2013), ‘Automated unsupervised classification of the sloan digital sky survey stellar spectra using k-means clustering’, *The Astrophysical Journal* **763**(1), 50.

Arias-Castro, E. & Pu, X. (2017), ‘A simple approach to sparse clustering’, *Computational Statistics & Data Analysis* **105**, 217–228.

Arthur, D. & Vassilvitskii, S. (2007), k-means++: The advantages of careful seeding, *in* ‘Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms’, Vol. 7, pp. 1027–1035.

Aschenbruck, R., Szepannek, G. & Wilhelm, A. F. (2023), ‘Imputation strategies for clustering mixed-type data with missing values’, *Journal of Classification* **40**(1), 2–24.

Audigier, V. & Niang, N. (2023), ‘Clustering with missing data: which equivalent for rubin’s rules?’, *Advances in Data Analysis and Classification* **17**(3), 623–657.

Bartlett, P. L. & Mendelson, S. (2002), ‘Rademacher and gaussian complexities: Risk bounds and structural results’, *Journal of Machine Learning Research* **3**(Nov), 463–482.

Becht, E., McInnes, L., Healy, J., Dutertre, C.-A., Kwok, I. W., Ng, L. G., Ginhoux, F. & Newell, E. W. (2019), ‘Dimensionality reduction for visualizing single-cell data using umap’, *Nature biotechnology* **37**(1), 38–44.

Bishop, C. M. (2006), *Pattern recognition and machine learning*, Springer.

Chakraborty, S. & Das, S. (2020), ‘Detecting meaningful clusters from high-dimensional data: A strongly consistent sparse center-based clustering approach’, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **44**(6), 2894–2908.

Chan, Y.-b. & Hall, P. (2010), 'Using evidence of mixed populations to select variables for clustering very high-dimensional data', *Journal of the American Statistical Association* **105**(490), 798–809.

Chang, X., Wang, Y., Li, R. & Xu, Z. (2018), 'Sparse k-means with l_∞/l_0 penalty for high-dimensional data clustering', *Statistica Sinica* **28**(3), 1265–1284.

Chi, J. T., Chi, E. C. & Baraniuk, R. G. (2016), 'k-pod: A method for k-means clustering of missing data', *The American Statistician* **70**(1), 91–99.

Dey, S., Das, S. & Mallipeddi, R. (2020), The sparse minmax k-means algorithm for high-dimensional clustering., *in* 'IJCAI', pp. 2103–2110.

Dhillon, I. S., Guan, Y. & Kulis, B. (2004), Kernel k-means: spectral clustering and normalized cuts, *in* 'Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining', pp. 551–556.

Ding, C. & He, X. (2004), K-means clustering via principal component analysis, *in* 'Proceedings of the twenty-first international conference on Machine learning', p. 29.

Ding, C., He, X. & Simon, H. D. (2005), On the equivalence of nonnegative matrix factorization and spectral clustering, *in* 'Proceedings of the 2005 SIAM international conference on data mining', SIAM, pp. 606–610.

Fan, J. & Li, R. (2001), 'Variable selection via nonconcave penalized likelihood and its oracle properties', *Journal of the American statistical Association* **96**(456), 1348–1360.

Fang, Y. & Wang, J. (2012), 'Selection of the number of clusters via the bootstrap method', *Computational Statistics & Data Analysis* **56**(3), 468–477.

Fraley, C. & Raftery, A. E. (2002), 'Model-based clustering, discriminant analysis, and density estimation', *Journal of the American statistical Association* **97**(458), 611–631.

Friedman, J. H. & Meulman, J. J. (2004), 'Clustering objects on subsets of attributes (with discussion)', *Journal of the Royal Statistical Society Series B: Statistical Methodology* **66**(4), 815–849.

Fukumizu, K. (2010), *Introduction to kernel method*, Asakura Publishing Co., Ltd.

Garreau, D., Jitkrittum, W. & Kanagawa, M. (2017), 'Large sample analysis of the median heuristic', *arXiv:1707.07269* .

Guan, X. & Terada, Y. (2023), 'Sparse kernel k-means for high-dimensional data', *Pattern Recognition* **144**, 109873.

Hartigan, J. A. & Wong, M. A. (1979), 'A k-means clustering algorithm', *Journal of the Royal Statistical Society. Series C (Applied Statistics)* **28**(1), 100–108.

Hathaway, R. J. & Bezdek, J. C. (2001), 'Fuzzy c-means clustering of incomplete data', *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* **31**(5), 735–744.

Hofmeyr, D. P. (2020), 'Degrees of freedom and model selection for k-means clustering', *Computational Statistics & Data Analysis* **149**, 106974.

Honaker, J., King, G. & Blackwell, M. (2011), 'Amelia ii: A program for missing data', *Journal of statistical software* **45**(7), 1–47.

Huang, J. Z., Ng, M. K., Rong, H. & Li, Z. (2005), 'Automated variable weighting in k-means type clustering', *IEEE transactions on pattern analysis and machine intelligence* **27**(5), 657–668.

Hunter, D. R. & Lange, K. (2004), 'A tutorial on mm algorithms', *The American Statistician* **58**(1), 30–37.

Jin, J. & Wang, W. (2016), 'Influential features pca for high dimensional clustering', *The Annals of Statistics* **44**(6), 2323–2359.

Kim, J. & Park, H. (2008), Sparse nonnegative matrix factorization for clustering, Technical report, Georgia Institute of Technology.

Kim, T., Chen, I. R., Lin, Y., Wang, A. Y.-Y., Yang, J. Y. H. & Yang, P. (2019), 'Impact of similarity metrics on single-cell rna-seq data clustering', *Briefings in bioinformatics* **20**(6), 2316–2326.

Kiselev, V. Y., Kirschner, K., Schaub, M. T., Andrews, T., Yiu, A., Chandra, T., Natarajan, K. N., Reik, W., Barahona, M., Green, A. R. et al. (2017), 'Sc3: consensus clustering of single-cell rna-seq data', *Nature methods* **14**(5), 483–486.

Le Morvan, M., Josse, J., Scornet, E. & Varoquaux, G. (2021), 'What's a good imputation to predict with missing values?', *Advances in Neural Information Processing Systems* **34**, 11530–11540.

Levrard, C. (2015), 'Nonasymptotic bounds for vector quantization in Hilbert spaces', *The Annals of Statistics* **43**(2), 592 – 619.

Levrard, C. (2018), 'Sparse oracle inequalities for variable selection via regularized quantization', *Bernoulli* **24**(1), 271 – 296.

Lithio, A. & Maitra, R. (2018), 'An efficient k-means-type algorithm for clustering datasets with incomplete records', *Statistical Analysis and Data Mining: The ASA Data Science Journal* **11**(6), 296–311.

Little, R. J. & Rubin, D. B. (2019), *Statistical analysis with missing data*, John Wiley & Sons, New York.

Lloyd, S. (1982), 'Least squares quantization in pcm', *IEEE Transactions on Information Theory* **28**(2), 129–137.

Maldonado, S., Carrizosa, E. & Weber, R. (2015), 'Kernel penalized k-means: A feature selection method based on kernel k-means', *Information sciences* **322**, 150–160.

McInnes, L., Healy, J. & Melville, J. (2018), 'Umap: Uniform manifold approximation and projection for dimension reduction', *arXiv preprint arXiv:1802.03426* .

Mohri, M., Rostamizadeh, A. & Talwalkar, A. (2018), *Foundations of machine learning*, MIT press.

Pan, W. & Shen, X. (2007), 'Penalized model-based clustering with application to variable selection.', *Journal of machine learning research* **8**(5).

Paul, D., Chakraborty, S., Das, S. & Xu, J. (2022), 'Implicit annealing in kernel spaces: A strongly consistent clustering approach', *IEEE Transactions on Pattern Analysis and Machine Intelligence* .

Qi, R., Ma, A., Ma, Q. & Zou, Q. (2020), 'Clustering and classification methods for single-cell rna-sequencing data', *Briefings in bioinformatics* **21**(4), 1196–1208.

Raymaekers, J. & Zamar, R. H. (2022), 'Regularized k-means through hard-thresholding', *Journal of Machine Learning Research* **23**(93), 1–48.

Sportisse, A., Boyer, C. & Josse, J. (2020), 'Imputation and low-rank estimation with missing not at random data', *Statistics and Computing* **30**(6), 1629–1643.

Sportisse, A., Marbac, M., Laporte, F., Celeux, G., Boyer, C., Josse, J. & Biernacki, C. (2024), 'Model-based clustering with missing not at random data', *Statistics and Computing* **34**(4), 135.

Steinhaus, H. et al. (1956), 'Sur la division des corps matériels en parties', *Bull. Acad. Polon. Sci (in French)* **1**(804), 801.

Steinwart, I. & Christmann, A. (2008), *Support vector machines*, Springer Science & Business Media.

Sun, W., Wang, J. & Fang, Y. (2012), 'Regularized k-means clustering of high-dimensional data and its asymptotic consistency', *Electronic Journal of Statistics* **6**, 148 – 167.

Terada, Y. & Guan, X. (2024), 'Some notes on the *k*-means clustering for missing data'.
URL: *https://arxiv.org/abs/2410.00546*

Terada, Y. & Yamamoto, M. (2019), Kernel normalized cut: A theoretical revisit, *in* 'International Conference on Machine Learning', PMLR, pp. 6206–6214.

Tibshirani, R., Walther, G. & Hastie, T. (2002), 'Estimating the Number of Clusters in a Data Set Via the Gap Statistic', *Journal of the Royal Statistical Society Series B: Statistical Methodology* **63**(2), 411–423.

Treutlein, B., Lee, Q. Y., Camp, J. G., Mall, M., Koh, W., Shariati, S. A. M., Sim, S., Neff, N. F., Skotheim, J. M., Wernig, M. et al. (2016), 'Dissecting direct reprogramming from fibroblast to neuron using single-cell rna-seq', *Nature* **534**(7607), 391–395.

Usoskin, D., Furlan, A., Islam, S., Abdo, H., Lönnerberg, P., Lou, D., Hjerling-Leffler, J., Haeggström, J., Kharchenko, O., Kharchenko, P. V. et al. (2015), 'Unbiased classification of sensory neuron types by large-scale single-cell rna sequencing', *Nature neuroscience* **18**(1), 145–153.

Van Buuren, S. & Groothuis-Oudshoorn, K. (2011), 'mice: Multivariate imputation by chained equations in r', *Journal of statistical software* **45**, 1–67.

Wagstaff, K. (2004), Clustering with missing values: No imputation required, *in* 'Classification, Clustering, and Data Mining Applications: Proceedings of the Meeting of the International Federation of Classification Societies (IFCS)', pp. 649–658.

Wang, B., Zhang, Y., Sun, W. W. & Fang, Y. (2018), 'Sparse convex clustering', *Journal of Computational and Graphical Statistics* **27**(2), 393–403.

Wang, J. (2010), 'Consistent selection of the number of clusters via crossvalidation', *Biometrika* **97**(4), 893–904.

Wang, S., Li, M., Hu, N., Zhu, E., Hu, J., Liu, X. & Yin, J. (2019), 'K-means clustering with incomplete data', *IEEE Access* **7**, 69162–69171.

Williams, C. & Seeger, M. (2000), 'Using the nyström method to speed up kernel machines', *Advances in neural information processing systems* **13**.

Witten, D. M. & Tibshirani, R. (2010), 'A framework for feature selection in clustering', *Journal of the American Statistical Association* **105**(490), 713–726.

Yang, M.-S. & Benjamin, J. B. (2023), 'Sparse possibilistic c-means clustering with lasso', *Pattern Recognition* **138**, 109348.

Yang, Y. & Zou, H. (2015), 'A fast unified algorithm for solving group-lasso penalize learning problems', *Statistics and Computing* **25**, 1129–1141.

Yuan, M. & Lin, Y. (2006), 'Model selection and estimation in regression with grouped variables', *Journal of the Royal Statistical Society Series B: Statistical Methodology* **68**(1), 49–67.

Zeng, H. & Cheung, Y.-M. (2009), 'A new feature selection method for gaussian mixture clustering', *Pattern Recognition* **42**(2), 243–250.

Zha, H., He, X., Ding, C., Gu, M. & Simon, H. (2001), 'Spectral relaxation for k-means clustering', *Advances in neural information processing systems* **14**.

Zhang, Z., Lange, K. & Xu, J. (2020), 'Simple and scalable sparse k-means clustering via feature ranking', *Advances in Neural Information Processing Systems* **33**, 10148–10160.