K-CDFs: a Nonparametric Clustering Algorithm via Cumulative Distribution Function

Jicai Liu*
School of Statistics and Mathematics,
Shanghai Lixin University of Accounting and Finance, Shanghai, China
Jinhong Li
School of Statistics, East China Normal University, Shanghai, China
Riquan Zhang
School of Statistics, East China Normal University, Shanghai, China

June 7, 2022

Abstract

We propose a novel partitioning clustering procedure based on the cumulative distribution function (CDF), called K-CDFs. For univariate data, the K-CDFs represent the cluster centers by empirical CDFs and assign each observation to the closest center measured by the Cramér-von Mises distance. The procedure is nonparametric and does not require assumptions on cluster distributions imposed by mixture models. A projection technique is used to generalize the K-CDFs for univariate data to an arbitrary dimension. The proposed procedure has several appealing properties. It is robust to heavy-tailed data, is not sensitive to the data dimensions, does not require moment conditions on data and can effectively detect linearly nonseparable clusters. To implement the K-CDFs, we propose two kinds of algorithms: a greedy algorithm as the classical Lloyd's algorithm and a spectral relaxation algorithm. We illustrate the finite sample performance of the proposed algorithms through simulation experiments and empirical analyses of several real datasets.

Keywords: Nonparametric partitioning clustering; K-means; spectral relaxation; projection mean variance; nonparametric ANOVA

*Corresponding author. liujicai1234@126.com.

1 Introduction

Clustering is a fundamental tool to identify homogenous groups of data points and various clustering methods have emerged during the past decade. Perhaps K-means clustering (Steinhaus, 1956; MacQueen, 1967) is the most well-studied method. It can be easily implemented by Lloyd's algorithm (Lloyd, 1982), which initializes the cluster center seeds, iteratively assigns the data to their closest center and updates the cluster centers. Because of its simplicity, K-means clustering is a very popular tool in exploratory data analysis and data mining.

Another class of widely used clustering methods is model-based in the statistics literature; see Fraley and Raftery (2002). It is assumed that the data distribution is a mixture of several basic parametric distributions. Then, each cluster can be modeled by each component of the mixture, and data points are allocated to the most likely component by the Bayes theorem. K-means can be viewed as a special case of Gaussian mixture models (GMM) by assuming Gaussian clusters with equal variance and partitioning data via a hard clustering scheme. Thus, the mixture models are more flexible than K-means.

A major restriction on mixture modeling is the parametric assumptions on cluster distributions in many applications. In this paper, we develop a new nonparametric clustering approach, called K-CDFs, which is similar to K-means but has nonparametric cluster centers. The K-CDFs do not require assumptions about cluster distributions imposed by the mixture modeling methods. A key technique of the K-CDFs is to employ the nonparametric variance component decomposition (ANOVA) proposed by Liu et al. (2022). The nonparametric ANOVA is analogous to the classical parametric ANOVA, except that the former does not rely on assumptions on the distribution of data. The K-CDFs use the within-cluster variation from the nonparametric ANOVA and search for the best partition by minimizing the variation. The K-CDFs procedure inherits all advantages of the nonparametric ANOVA; for example, it is not sensitive to the dimensions of data and does not

require moment conditions on data.

It is well known that K-means cannot effectively separate clusters that are nonlinearly separated. To remedy the issue, K-means was extended to a kernel version using the kernel trick. See, Schölkopf et al. (1998); Dhillon et al. (2004); Filippone et al. (2008). The kernel K-means embeds the data into a higher-dimensional space via nonlinear mapping and then partitions the points by K-means in the new space. Although the kernel K-means can detect linearly nonseparable clusters, there are certain drawbacks; for example, it depends heavily on the choice of the kernel and suffers from a lack of interpretability due to nonlinear mapping. However, our K-CDFs are free of tuning parameters and have a good statistical interpretation while retaining the efficiency of the kernel K-means for nonlinearly separable data.

Along the lines of the K-means algorithm, we propose a greedy algorithm as Lloyd's algorithm, to implement the K-CDFs. The classical Lloyd's algorithm is prone to local minima, as are our Lloyd-type K-CDFs. To address the problem, we equivalently formulate the K-CDFs procedure as a trace maximization problem, which is a nonconvex optimization problem. We propose a spectral relaxation method to optimize it in two steps: (1) we relax the problem into a tractable spectral analysis problem by replacing the binary assignment matrix with a continuous-valued matrix, and (2) we use a rounding scheme to obtain a partition from the continuous relaxation. The simulation and real data results show that the spectral relaxation K-CDF algorithm can avoid local optima and is comparable to existing counterparts, such as K-means, GMM, kernel K-means and energy statistic-based clustering (Li and Rizzo, 2017; Franca et al., 2021).

We summarize the main contributions of this paper as follows:

(a) A new nonparametric statistical clustering algorithm, the K-CDFs, is proposed by the nonparametric ANOVA, and a Lloyd-type algorithm is developed to implement the K-CDFs;

(b) A spectral relaxation algorithm is developed to avoid local optima from which the Lloyd-type K-CDF algorithm suffers;

(c) We provide an equivalent and weighted formulation for the K-CDFs and establish connections with existing clustering methods, such as kernel K-means and energy statistic-based clustering.

The rest of the paper is organized as follows. In Section 2, we introduce the K- CDFs clustering procedure and propose the Lloyd-type algorithm to implement the K-CDFs. In Section 3, we introduce the spectral relaxation algorithm. Simulation experiments and real data analysis are presented in Sections 4 and 5, respectively. We complete the paper with a brief discussion in Section 6. All technical proofs are provided in the Supplementary file.

2 Methodology

Let $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ with $\mathbf{x}_i \in \mathcal{R}^p$ be the data points. We assume that $\mathbf{x}_i$ is continuous, although the procedure extends directly to discrete data. Our goal is to partition the data into K disjoint clusters such that each data point belongs to only one cluster. Here, the number of clusters K is prespecified. Let $V = \{1, 2, \ldots, n\}$ be the index set of the data. Equivalently, the clustering problem partitions V into K subsets $\{V_k\}_{k=1}^K$, that is, $\cup_{k=1}^K V_k = V$ and $V_k \cap V_j = \emptyset, k \neq j$, with the cardinality $\#\{V_k\} = n_k$ and $\sum_{k=1}^K n_k = n$. Denote the cluster assignment matrix $\mathbf{E} = [\mathbf{e}_1, \cdots, \mathbf{e}_n]^T \in \mathcal{R}^{n \times K}$, where $\mathbf{e}_i \in \{0, 1\}^{K \times 1}$ is a binary vector, whose k-th element is 1 if only and if the index of $\mathbf{x}_i$ belongs to V_k.

2.1 K-CDFs for univariate data

To illustrate our methods, we first introduce the K-CDFs for univariate data. In the next section, we generalize the procedure to an arbitrary dimension by integrating over all one-dimensional projections.

The classical K-means problem can be formulated as the minimization of the following objective function

$$\widetilde{L}(\mathscr{V}, \widetilde{\Theta}) = \frac{1}{n} \sum_{k=1}^{K} \sum_{i \in V_k} (x_i - \theta_k)^2,$$

with respect to $\mathscr{V} = \{V_1, \ldots, V_K\}$ and $\widetilde{\Theta} = \{\theta_1, \ldots, \theta_K\}$. It can be seen that $\widetilde{L}(\mathscr{V}, \widetilde{\Theta})$ is the variance within the clusters for given $\mathscr{V}$, if θ_k is within the cluster mean. Motivated by the nonparametric ANOVA in Liu et al. (2022), we propose a nonparametric version of $\widetilde{L}(\mathscr{V}, \widetilde{\Theta})$, given by

$$L(\mathscr{V}, \Theta) = \frac{1}{n} \sum_{k=1}^{K} \sum_{i \in V_k} \int_{\mathcal{R}} [I(x_i \leq x) - \theta_k(x)]^2 d\widehat{F}(x),$$

where $\Theta = \Theta(x) = \{\theta_1(x), \ldots, \theta_K(x)\}$ and $\widehat{F}(x)$ is the empirical CDF of $\{x_i\}_{i=1}^n$. From Liu et al. (2022), $L(\mathscr{V}, \Theta)$ can measure the within-cluster variation if $\theta_k(x)$ is the empirical CDF of $\{x_i, i \in V_k\}$. Different from $\widetilde{L}(\mathscr{V}, \widetilde{\Theta})$, $L(\mathscr{V}, \Theta)$ enjoys a model-free property that does not rely on assumptions on the data distribution. It is noteworthy that $L(\mathscr{V}, \Theta)$ is not exactly equal to the within-cluster variation in Liu et al. (2022) due to the unknown $\mathscr{V}$ and Θ.

Similar to K-means, we can minimize $L(\mathscr{V}, \Theta)$ with respect to $\mathscr{V}$ and Θ to perform clustering. Therefore, a coordinate descend method can be used. Specifically, by simple calculation, for fixed $\{V_k\}_{k=1}^K$, the optimal centroids can be achieved by

$$\widehat{\theta}_k(x) = \widehat{F}_k(x) = \frac{1}{n_k} \sum_{i \in V_k} I(x_i \leq x), k = 1, \ldots, K. \tag{1}$$

Plugging $\widehat{\Theta} = \{\widehat{\theta}_1(x), \ldots, \widehat{\theta}_K(x)\}$ into $L(\mathscr{V}, \Theta)$, we can obtain the estimator of $\mathscr{V}$ by minimizing the following objective function:

$$L(\mathscr{V}, \widehat{\Theta}) = \frac{1}{n} \sum_{k=1}^{K} \sum_{i \in V_k} \int_{\mathcal{R}} [I(x_i \leq x) - \widehat{F}_k(x)]^2 d\widehat{F}(x). \tag{2}$$

By the definition of the binary assignment matrix $\mathbf{E}$, we have $V_k = \{i \in \{1, \ldots, n\} : \mathbf{E}_{ik} = 1\}$. Thus, we obtain that

$$\frac{1}{n} \sum_{k=1}^{K} \sum_{i \in V_k} \int_{\mathcal{R}} [I(x_i \leq x) - \widehat{F}_k(x)]^2 d\widehat{F}(x) = \frac{1}{n} \sum_{k=1}^{K} \sum_{i=1}^{n} \mathbf{E}_{ik} \int_{\mathcal{R}} [I(x_i \leq x) - \widehat{F}_k(x)]^2 d\widehat{F}(x),$$

which implies that minimizing (2) is equivalent to the optimization problem

$$\min_{\mathbf{E}} \frac{1}{n} \sum_{i=1}^{n} \sum_{k=1}^{K} \mathbf{E}_{ik} \int_{\mathcal{R}} [I(x_i \leq x) - \widehat{F}_k(x)]^2 d\widehat{F}(x). \tag{3}$$

Note that, for given $i \in \{1, \cdots, n\}$, we have that

$$\sum_{k=1}^{K} \mathbf{E}_{ik} \int_{\mathcal{R}} [I(x_i \leq x) - \widehat{F}_k(x)]^2 d\widehat{F}(x) \geq \min_{1 \leq j \leq K} \int_{\mathcal{R}} [I(x_i \leq x) - \widehat{F}_j(x)]^2 d\widehat{F}(x).$$

The equality holds if only if $\mathbf{E}_{ik} = 1$ for whichever value of k gives the minimum value of $\int_{\mathcal{R}} [I(x_i \leq x) - \widehat{F}_j(x)]^2 d\widehat{F}(x)$. That is, the minimizer of the objective function (3) can be given by

$$\mathbf{E}_{ik}^{\mathtt{opt}} = \begin{cases} 1, & \text{if } k = \arg\min\limits_{1 \leq j \leq K} \int_{\mathcal{R}} [I(x_i \leq x) - \widehat{F}_j(x)]^2 d\widehat{F}(x), \\ 0, & \text{otherwise}, \end{cases} \tag{4}$$

for $i \in \{1, \cdots, n\}$ and $k \in \{1, \cdots, K\}$. Thus, the optimal partition can be estimated by $\widehat{V}_k = \{i \in \{1, \ldots, n\} : \mathbf{E}_{ik}^{\mathtt{opt}} = 1\}$.

Iterating the above two formulas in (1) and (4), we can cluster the data $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$. Note that each optimal centroid $\widehat{\theta}_k(x)$ in (1) is the empirical CDF of the cluster $\{x_i, i \in V_k\}$. Moreover, it can be seen in (4) that each observation is allocated to the closest centroid based on the distance $\int_{\mathcal{R}} [I(x_i \leq x) - \widehat{F}_j(x)]^2 d\widehat{F}(x)$, which is a Cramér-von Mises distance between a special CDF, the indicator function $I(x_i \leq x)$, and the empirical CDF. The procedure works very similarly to K-means, except that the centroid and distance are different. Thus, we call our procedure K-CDFs.

According to the classical Lloyd's algorithm for K-means, we can develop a Lloyd-type K-CDF algorithm in Algorithm 1 to implement the above procedure. Algorithm 1 always converges because it keeps decreasing the value of $L(\mathscr{V}, \Theta)$. However, note that the algorithm may converge to a local minimum or even a saddle. In Section 3, we provide a strategy to avoid local minima.

Algorithm 1 Lloyd-type K-CDF algorithm for univariate data

1: Let $t = 0$. Randomly assign a number, from 1 to K, to each observation and obtain the initial assignment matrix $\mathbf{E}^{(t)}$;

2: Repeat the following steps until convergence:

(a) For each of the K clusters, compute the cluster centroid

$$\widehat{F}_k^{(t)}(x) = \frac{1}{n_k^{(t)}} \sum_{i \in V_k^{(t)}} I(x_i \leq x), k = 1, 2, \ldots K,$$

where $V_k^{(t)} = \{i \in \{1, 2, \ldots, n\} : \mathbf{E}_{ik}^{(t)} = 1\}$ and $n_k^{(t)} = \#V_k^{(t)}$;

(b) Update the assignment matrix by

$$\mathbf{E}_{ik}^{(t+1)} = \begin{cases} 1, & \text{if } \; k = \underset{1 \leq j \leq K}{\arg\min} \int_{\mathcal{R}} [I(x_i \leq x) - \widehat{F}_j^{(t)}(x)]^2 d\widehat{F}(x), \\ 0, & \text{otherwise.} \end{cases}$$

Set $t \leftarrow t + 1$;

3: Output $\mathbf{E}^{(t)}$.

2.2 K-CDFs for multivariate data

Note that Algorithm 1 is substantially a rank-based method and thus may be unstable when the dimension of $\mathbf{x}_i$ is high. To overcome this shortcoming, we apply the projection technique (Escanciano, 2006; Zhu et al., 2017; Kim et al., 2020; Liu et al., 2022) and extend the K-CDFs for univariate data to an arbitrary dimension.

Let $c_p = \pi^{p/2-1}/\Gamma(p/2)$ and $\Gamma(\cdot)$ be the gamma function. Assume that $\mathbb{S}^{p-1} = \{\beta \in \mathcal{R}^p : \|\beta\| = 1\}$ is the unit hypersphere in $\mathcal{R}^p$ for any $p > 1$. Using the projection data $\{\beta^T\mathbf{x}_i, \beta \in \mathbb{S}^{p-1}\}_{i=1}^n$, a multivariate version of $L(\mathscr{V}, \Theta)$ can be defined as

$$L_P(\mathscr{V}, \Theta) = \frac{1}{nc_p} \sum_{k=1}^{K} \sum_{i \in V_k} \int_{\mathbb{S}^{p-1}} \int_{\mathcal{R}} [I(\beta^T\mathbf{x}_i \le u) - \theta_k(u)]^2 d\widehat{F}_\beta(u) d\beta, \tag{5}$$

where $\widehat{F}_\beta(u) = \frac{1}{n}\sum_{i=1}^n I(\beta^T\mathbf{x}_i \le u)$.

To minimize $L_P(\mathscr{V}, \Theta)$ with respect to $\mathscr{V}$ and Θ, we can also use the coordinate descending method in the above section. For a fixed $\mathscr{V}$, the minimizer of $L_P(\mathscr{V}, \Theta)$ with respect to Θ is

$$\widehat{\theta}_k(u) = \widehat{F}_{k,\beta}(u) = \frac{1}{n_k} \sum_{i \in V_k} I(\beta^T\mathbf{x}_i \le u). \tag{6}$$

Similar to the centroids in (1), $\widehat{\theta}_k(u)$ is also an empirical CDF for a given β. Plugging $\widehat{\theta}_k(u)$ into $L_P(\mathscr{V}, \Theta)$, we can obtain the optimal $\mathscr{V}$ by optimizing the following problem:

$$\min_{\mathscr{V}} \frac{1}{nc_p} \sum_{k=1}^{K} \sum_{i \in V_k} \int_{\mathbb{S}^{p-1}} \int_{\mathcal{R}} [I(\beta^T\mathbf{x}_i \le u) - \widehat{F}_{k,\beta}(u)]^2 d\widehat{F}_\beta(u) d\beta. \tag{7}$$

By a similar argument of (3) and (4), we obtain that (7) is equivalent to

$$\min_{\mathbf{E}} \frac{1}{nc_p} \sum_{i=1}^{n} \sum_{k=1}^{K} \mathbf{E}_{ik} \int_{\mathbb{S}^{p-1}} \int_{\mathcal{R}} [I(\beta^T\mathbf{x}_i \le u) - \widehat{F}_{k,\beta}(u)]^2 d\widehat{F}_\beta(u) d\beta, \tag{8}$$

and the (8) minimizer satisfies

$$\mathbf{E}_{ik}^{\mathsf{opt}} = \begin{cases} 1, & \text{if } k = \underset{1 \le j \le K}{\arg\min} \int_{\mathbb{S}^{p-1}} \int_{\mathcal{R}} [I(\beta^T\mathbf{x}_i \le u) - \widehat{F}_{j,\beta}(u)]^2 d\widehat{F}_\beta(u) d\beta, \\ 0, & \text{otherwise,} \end{cases} \tag{9}$$

for $i \in \{1, \cdots, n\}$ and $k \in \{1, \cdots, K\}$. The optimal partition can be given by $\widehat{V}_k = \{i \in \{1, \ldots, n\} : \mathbf{E}_{ik}^{\mathtt{opt}} = 1\}$.

Note that $\mathbf{E}_{ik}^{\mathtt{opt}}$ in (9) involves integration over the p-dimensional unit sphere, which can be computationally costly. Fortunately, the following lemma ensures that it has closed form.

Lemma 1. *(Escanciano, 2006) For any nonzero vectors $\boldsymbol{v}_1, \boldsymbol{v}_2 \in \mathcal{R}^p$, we have*

$$\int_{\mathbb{S}^{p-1}} I(\beta^T \mathbf{v}_1 \leq 0) I(\beta^T \mathbf{v}_2 \leq 0) d\beta = c_p \Big[\pi - \arccos\Big\{\frac{\boldsymbol{v}_1^T \boldsymbol{v}_2}{\|\boldsymbol{v}_1\|_2 \|\boldsymbol{v}_2\|_2}\Big\}\Big].$$

By Lemma 1, the following results provide an equivalent and computationally feasible form for $\mathbf{E}_{ik}^{\mathtt{opt}}$ in (9).

Theorem 1. *For any partition $\{V_k\}_{k=1}^K$, let $\mathbf{D} = \mathrm{diag}(n_1, \cdots, n_K)$, where n_k is the cardinality of V_k. For any point $\mathbf{x}_i$, we have that*

$$\frac{1}{c_p} \int_{\mathbb{S}^{p-1}} \int_{\mathcal{R}} [I(\beta^T \mathbf{x}_i \leq u) - \widehat{F}_{k,\beta}(u)]^2 d\widehat{F}_\beta(u) d\beta = -[\mathbf{e}_{i,n} - \mathbf{E}\mathbf{D}^{-1}\mathbf{e}_{k,K}]^T \boldsymbol{K}_P [\mathbf{e}_{i,n} - \mathbf{E}\mathbf{D}^{-1}\mathbf{e}_{k,K}],$$

where the (i,j)-element of $\boldsymbol{K}_P \in \mathcal{R}^{n \times n}$ is equal to

$$\Big(\boldsymbol{K}_P\Big)_{ij} = \frac{1}{n} \sum_{k=1}^{n} \arccos\left(\frac{(\mathbf{x}_i - \mathbf{x}_k)^T (\mathbf{x}_j - \mathbf{x}_k)}{\|\mathbf{x}_i - \mathbf{x}_k\|_2 \|\mathbf{x}_j - \mathbf{x}_k\|_2}\right),$$

$\mathbf{e}_{i,n} \in \mathcal{R}^{n \times 1}$ is a binary vector whose i-th entry is 1 while other entries are 0, and $\mathbf{e}_{k,K} \in \mathcal{R}^{K \times 1}$ is defined in the same way.

Note that the cluster centers $\widehat{\theta}_k(u) = \widehat{F}_{k,\beta}(u)$ in (6) depend on the projection direction $\beta \in \mathbb{S}^{p-1}$, which lacks interpretability in practice, unlike those in (1). However, the main task of a clustering algorithm is to estimate the cluster assignment matrix $\mathbf{E}$ or equivalently the partition $\mathscr{V}$ rather than the cluster centers. Thus, using Theorem 1 and Algorithm 1, we can propose a modified Lloyd-type K-CDF algorithm for multivariate data, listed in Algorithm 2. Furthermore, the following theorem provides a connection between Algorithms 1 and 2.

Algorithm 2 Lloyd-type K-CDF algorithm for multivariate data

1: Let $t = 0$. Randomly assign a number, from 1 to K, to each observation and obtain the initial assignment matrix $\mathbf{E}^{(t)}$;

2: Repeat the following steps until convergence:

(a) Calculate the matrix $\mathbf{D}^{(t)} = \text{diag}(n_1^{(t)}, \cdots, n_K^{(t)})$, where $n_k^{(t)} = \#\{i \in \{1, 2, \ldots, n\} : \mathbf{E}_{ik}^{(t)} = 1\}$;

(b) Update the assignment matrix by

$$\mathbf{E}_{ik}^{(t+1)} = \begin{cases} 1, & \text{if } k = \underset{1 \le j \le K}{\arg\min} -[\mathbf{e}_{i,n} - \mathbf{E}^{(t)}\{\mathbf{D}^{(t)}\}^{-1}\mathbf{e}_{j,K}]^T \mathbf{K}_{\mathrm{P}} [\mathbf{e}_{i,n} - \mathbf{E}^{(t)}\{\mathbf{D}^{(t)}\}^{-1}\mathbf{e}_{j,K}], \\ 0, & \text{otherwise.} \end{cases}$$

Set $t \leftarrow t + 1$;

3: Output $\mathbf{E}^{(t)}$.

Theorem 2. *Assume that each* $\mathbf{x}_i$ *is univariate. Then, we have that*

(i) $\int_{\mathcal{R}} [I(\mathbf{x}_i \le x) - \widehat{F}_k(x)]^2 d\widehat{F}(x) = [\mathbf{e}_{i,n} - \mathbf{E}\mathbf{D}^{-1}\mathbf{e}_{k,K}]^T \boldsymbol{K} [\mathbf{e}_{i,n} - \mathbf{E}\mathbf{D}^{-1}\mathbf{e}_{k,K}]$, *where the* (i, j)*-element of* $\boldsymbol{K} \in \mathcal{R}^{n \times n}$ *is equal to* $(\boldsymbol{K})_{ij} = \frac{1}{n}\sum_{k=1}^{n} I(\mathbf{x}_i \le \mathbf{x}_k) I(\mathbf{x}_j \le \mathbf{x}_k)$;

(ii) $\frac{1}{c_p}\int_{\mathbb{S}^{p-1}} \int_{\mathcal{R}} [I(\beta^T \mathbf{x}_i \le u) - \widehat{F}_{k,\beta}(u)]^2 d\widehat{F}_\beta(u) d\beta = 2\pi \int_{\mathcal{R}} [I(\mathbf{x}_i \le x) - \widehat{F}_k(x)]^2 d\widehat{F}(x)$, *with probability 1,*

where $\mathbf{e}_{i,n} \in \mathcal{R}^{n \times 1}$ *and* $\mathbf{e}_{k,K} \in \mathcal{R}^{K \times 1}$ *are defined in Theorem 1.*

As a byproduct, property (i) can be used to compute $\mathbf{E}_{ik}^{(t+1)}$ in Algorithm 1 in matrix form. When $\mathbf{x}_i$ is univariate, property (ii) shows that the optimal assignment matrix in (4) is equal to that in (9). That is, Algorithm 2 is exactly equivalent to Algorithm 1 for univariate data. Thus, we can generalize Algorithm 2 to an arbitrary dimension, although the objective function in (5) requires $p > 1$.

In practice, an important problem is that K needs to be pre-specified for the above K-CDFs. For K-means, the commonly used approaches to determine the optimal K include

the method based on prior information about data, the elbow method (Thorndike, 1953), the AIC or BIC criterion (Pelleg and Moore, 2000), the gap statistic (Tibshirani et al., 2001) among others. We can extend these approaches to the K-CDFs.

In the following theorem, we show that Algorithm 2 converges to a local optimum.

Theorem 3. *Algorithm 2 monotonically decreases the objective function in (8) until local convergence.*

In the Supplementary Material, we further establish the consistency of the K-CDFs in Theorem B.1, which shows that the estimated centers of the K-CDFs converge almost surely to the true centers. The result is parallel to that for the K-means algorithm in Pollard (1981). It is noteworthy that Theorem B.1 does not require any moment condition on the data, whereas K-means needs the second-moment condition $\int \|\mathbf{X}\|^2 dP < \infty$. Thus, if the moment condition is violated or when the data have heavy tails, K-means may lose efficiency, but the K-CDFs still exhibit powerful performance, which can be demonstrated by the numerical simulations.

3 Spectral relaxation for K-CDFs

Note that Algorithm 2 is a Lloyd-type algorithm and may suffer from the drawback that the classical Lloyd's algorithm is prone to become stuck at local minima. A strategy to avoid local minima is to search for good initializers such as K-means++. In the section, we adopt a different method proposed by Zha et al. (2002), convert the sum-of-squares minimization in (8) into an equivalent trace maximization problem and optimize it by a spectral relaxation and rounding method proposed by Zhang and Jordan (2008).

3.1 Equivalent form for K-CDFs

In this section, we obtain an ANOVA decomposition in matrix-vector form, which can be viewed as a generalization of Liu et al. (2022)'s results to the unsupervised clustering problem. For any given partition $\{V_k\}_{k=1}^K$ of $V = \{1, \ldots, n\}$, we define

$$\begin{aligned}
\mathrm{S}_T^{\mathsf{P}} &= \frac{1}{nc_p} \sum_{i=1}^{n} \int_{\mathbb{S}^{p-1}} \int_{\mathcal{R}} [I(\beta^T \mathbf{x}_i \le u) - \widehat{F}_\beta(u)]^2 d\widehat{F}_\beta(u) d\beta, \\
S_B^{\mathsf{P}} &= \frac{1}{nc_p} \sum_{k=1}^{K} n_k \int_{\mathbb{S}^{p-1}} \int_{\mathcal{R}} [\widehat{F}_{k,\beta}(u) - \widehat{F}_\beta(u)]^2 d\widehat{F}_\beta(u) d\beta, \\
\mathrm{S}_W^{\mathsf{P}} &= \frac{1}{nc_p} \sum_{k=1}^{K} \sum_{i \in V_k} \int_{\mathbb{S}^{p-1}} \int_{\mathcal{R}} [I(\beta^T \mathbf{x}_i \le u) - \widehat{F}_{k,\beta}(u)]^2 d\widehat{F}_\beta(u) d\beta,
\end{aligned}$$

where $\widehat{F}_{k,\beta}(u)$ is the empirical CDF of subset $\{\beta^T \mathbf{x}_i, i \in V_k\}$.

Note that the optimization problem in (7) can be rewritten as $\min_{\mathscr{V}} \mathrm{S}_W^{\mathsf{P}}$. The following result shows that $\min_{\mathscr{V}} \mathrm{S}_W^{\mathsf{P}}$ is equivalent to maximizing $\mathrm{S}_B^{\mathsf{P}}$.

Theorem 4. **(i)** *For any partition $\{V_k\}_{k=1}^K$, we have*

$$\mathrm{S}_T^P = \mathrm{S}_W^P + \mathrm{S}_B^P; \tag{10}$$

(ii) *S_T^P and S_B^P can be expressed in terms of matrix as*

$$\mathrm{S}_T^P = -\frac{1}{n} \operatorname{tr}(\mathbf{H}\boldsymbol{K}_P) \quad \text{and} \quad \mathrm{S}_B^P = -\frac{1}{n} \operatorname{tr}\left(\mathbf{E}^T \mathbf{H} \boldsymbol{K}_P \mathbf{H} \mathbf{E} \mathbf{D}^{-1}\right),$$

where $\mathbf{H} = \mathbf{I}_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T$, $\mathbf{I}_n$ is an $n \times n$ identity matrix, $\mathbf{1}_n$ is an $n \times 1$ vector of ones and $\boldsymbol{K}_P$ is defined in Theorem 1.

Theorem 4(i) provides an analog to the ANOVA decomposition, which is consistent with Liu et al. (2022)'s results for the multisample problem. By the decomposition in (10), we obtain that the minimization problem in (7) is equivalent to

$$\max_{\mathscr{V}} \mathrm{S}_B^{\mathsf{P}} = \max_{\mathscr{V}} \frac{1}{nc_p} \sum_{k=1}^{K} n_k \int_{\mathbb{S}^{p-1}} \int_{\mathcal{R}} [\widehat{F}_{k,\beta}(u) - \widehat{F}_\beta(u)]^2 d\widehat{F}_\beta(u) d\beta,$$

where $\mathscr{V} = \{V_1, \ldots, V_K\}$. By Theorem 4(ii), the above problem is equivalent to

$$\max_{\mathbf{E}} \left(S_B^{\mathsf{P}}\right) = \max_{\mathbf{E}} \left\{ -\frac{1}{n} \operatorname{tr}\left(\mathbf{E}^T \mathbf{H} \mathbf{K}_{\mathsf{P}} \mathbf{H} \mathbf{E} \mathbf{D}^{-1}\right) \right\}. \tag{11}$$

Similar to the classical mincut clustering approach (Zhang and Jordan, 2008), we can further develop a weighted version of the problem (11) by introducing weights associated with data points. Specifically, we define the weighted versions of $\mathrm{S}_T^{\mathsf{P}}$, S_W^{P} and $\mathrm{S}_B^{\mathsf{P}}$ as

$$\begin{aligned}
S_{T,\pi}^{\mathsf{P}} &= \frac{1}{c_p \sum_{i=1}^n \pi_i} \sum_{i=1}^n \pi_i \int_{\mathbb{S}^{p-1}} \int_{\mathcal{R}} [I(\beta^T \mathbf{x}_i \le u) - \widehat{F}_\beta^\pi(u)]^2 d\widehat{F}_\beta^\pi(u) d\beta, \\
S_{W,\pi}^{\mathsf{P}} &= \frac{1}{c_p \sum_{i=1}^n \pi_i} \sum_{k=1}^K \sum_{i \in V_k} \pi_i \int_{\mathbb{S}^{p-1}} \int_{\mathcal{R}} [I(\beta^T \mathbf{x}_i \le u) - \widehat{F}_{k,\beta}^\pi(u)]^2 d\widehat{F}_\beta^\pi(u) d\beta, \\
S_{B,\pi}^{\mathsf{P}} &= \frac{1}{c_p \sum_{i=1}^n \pi_i} \sum_{k=1}^K \sum_{i \in V_k} \pi_i \int_{\mathbb{S}^{p-1}} \int_{\mathcal{R}} [\widehat{F}_{k,\beta}^\pi(u) - \widehat{F}_\beta^\pi(u)]^2 d\widehat{F}_\beta^\pi(u) d\beta,
\end{aligned}$$

where $\boldsymbol{\pi} = [\pi_1, \ldots, \pi_n]^T$ is a user-defined weight vector with $\pi_i > 0$ and

$$\widehat{F}_\beta^\pi(u) = \frac{1}{\sum_{i=1}^n \pi_i} \sum_{i=1}^n \pi_i I(\beta^T \mathbf{x}_i \le u) \text{ and } \widehat{F}_{k,\beta}^\pi(u) = \frac{1}{\sum_{i \in V_k} \pi_i} \sum_{i \in V_k} \pi_i I(\beta^T \mathbf{x}_i \le u).$$

For the classical mincut clustering approach, $\boldsymbol{\pi} = [1, \cdots, 1]^T$ and $\boldsymbol{\pi} = [n_1, \cdots, n_K]^T$ are two most common weight vectors, which yield that the ratio cut clustering algorithm (Chan et al., 1994) and the normalized cut clustering algorithm (Shi and Malik, 2000). In practice, we can use the two weight vectors in $\mathrm{S}_T^{\mathsf{P}}$, S_W^{P} and $\mathrm{S}_B^{\mathsf{P}}$.

Let $\boldsymbol{\Pi} = \operatorname{diag}(\pi_1, \ldots, \pi_n)$ and $\mathbf{H}_\pi = \mathbf{I}_n - \frac{1}{\boldsymbol{\pi}^T \mathbf{1}_n} \boldsymbol{\pi} \mathbf{1}_n^T$. Using Lemma 1 and a similar argument of Theorem 4, we have the following results.

Theorem 5. **(i)** *We obtain the decomposition:* $S_{T,\pi}^P = S_{W,\pi}^P + S_{B,\pi}^P$;

(ii) $S_{T,\pi}^P$ *and* $S_{B,\pi}^P$ *can be expressed in terms of matrix as*

$$\begin{aligned}
S_{T,\pi}^P &= -\frac{1}{\boldsymbol{\pi}^T \mathbf{1}_n} \operatorname{tr}\left(\mathbf{H}_\pi \boldsymbol{\Pi} \mathbf{H}_\pi^T \boldsymbol{K}_P^\pi\right), \\
S_{B,\pi}^P &= -\frac{1}{\boldsymbol{\pi}^T \mathbf{1}_n} \operatorname{tr}\left(\mathbf{E}^T \boldsymbol{\Pi} \mathbf{H}_\pi^T \boldsymbol{K}_P^\pi \mathbf{H}_\pi \boldsymbol{\Pi} \mathbf{E} (\mathbf{E}^T \boldsymbol{\Pi} \mathbf{E})^{-1}\right),
\end{aligned}$$

where the (i,j)*-element of* $\boldsymbol{K}_P^\pi \in \mathcal{R}^{n \times n}$ *is equal to*

$$\left(\boldsymbol{K}_P^\pi\right)_{ij} = \frac{1}{\sum_{k=1}^n \pi_k} \sum_{k=1}^n \pi_k \arccos\left(\frac{(\mathbf{x}_i - \mathbf{x}_k)^T (\mathbf{x}_j - \mathbf{x}_k)}{\|\mathbf{x}_i - \mathbf{x}_k\|_2 \|\mathbf{x}_j - \mathbf{x}_k\|_2}\right).$$

Taking $\boldsymbol{\pi} = [1, \cdots, 1]^T$, we can see that Theorem 5 is equivalent to Theorem 4. By Theorem 5, the weighted version of the optimization problem (11) can be given by

$$\max_{\mathbf{E}} \left(S^{\mathsf{P}}_{B,\pi}\right) = \max_{\mathbf{E}} \left\{ - \operatorname{tr}\left(\mathbf{E}^T \boldsymbol{\Pi} \mathbf{H}_\pi^T \mathbf{K}_{\mathsf{P}}^\pi \mathbf{H}_\pi \boldsymbol{\Pi} \mathbf{E} (\mathbf{E}^T \boldsymbol{\Pi} \mathbf{E})^{-1}\right)\right\}. \tag{12}$$

Note that the optimization problem (12) (also (11)) is nonconvex and NP-hard; thus, a direct approach is computationally problematic. A natural idea is to optimize it by some relaxed algorithm. We present the algorithms in the following section.

3.2 Spectral relaxation for K-CDFs

In this section, we develop a spectral relaxation method to optimize the problem (12), where the problem (11) can be similarly solved. Our approach relaxes the binary assignment matrix $\mathbf{E} \in \mathcal{R}^{n \times K}$ into a continuous-valued matrix $\mathbf{Y} \in \mathcal{R}^{n \times (K-1)}$, which relies on the following lemma.

Lemma 2. *Let* $\mathbf{Y}$ *be an* $n \times (K-1)$ *real matrix such that: (a) the columns of* $\mathbf{Y}$ *are piecewise constant with respect to* $\mathbf{E}$*, i.e.* $\mathbf{Y} = \mathbf{E}\Phi$*, where* Φ *is a* $K \times (K-1)$ *matrix; (b)* $\mathbf{Y}^T \boldsymbol{\Pi} \mathbf{Y} = \mathbf{I}_{K-1}$*; (c)* $\mathbf{Y}^T \boldsymbol{\Pi} \mathbf{1}_n = \mathbf{0}$*. Then,* $\operatorname{tr}\left(\mathbf{E}^T \boldsymbol{\Pi} \mathbf{H}_\pi^T \boldsymbol{K}_P^\pi \mathbf{H}_\pi \boldsymbol{\Pi} \mathbf{E} (\mathbf{E}^T \boldsymbol{\Pi} \mathbf{E})^{-1}\right)$ *in* (12) *is equal to* $\operatorname{tr}\left(\mathbf{Y}^T \boldsymbol{\Pi} \mathbf{H}_\pi^T \boldsymbol{K}_P^\pi \mathbf{H}_\pi \boldsymbol{\Pi} \mathbf{Y}\right)$.

The above lemma can be derived by Proposition 1 in Zhang and Jordan (2008), where a simple proof is given in the Appendix. By dropping condition (a), we can formulate a relaxed version of the problem (12), given by

$$\begin{aligned} &\max_{\mathbf{Y} \in \mathcal{R}^{n \times (K-1)}} \operatorname{tr}\left(\mathbf{Y}^T \boldsymbol{\Pi} \mathbf{H}_\pi^T [-\mathbf{K}_{\mathsf{P}}^\pi] \mathbf{H}_\pi \boldsymbol{\Pi} \mathbf{Y}\right) \\ &\text{s.t.} \quad \mathbf{Y}^T \boldsymbol{\Pi} \mathbf{Y} = \mathbf{I}_{K-1} \text{ and } \mathbf{Y}^T \boldsymbol{\Pi} \mathbf{1}_n = \mathbf{0}. \end{aligned} \tag{13}$$

Letting $\mathbf{Y}_0 = \boldsymbol{\Pi}^{1/2} \mathbf{Y}$, we transform the above problem into the following one:

$$\begin{aligned} &\max_{\mathbf{Y}_0 \in \mathcal{R}^{n \times (K-1)}} \operatorname{tr}\left(\mathbf{Y}_0^T \boldsymbol{\Pi}^{1/2} \mathbf{H}_\pi^T [-\mathbf{K}_{\mathsf{P}}^\pi] \mathbf{H}_\pi \boldsymbol{\Pi}^{1/2} \mathbf{Y}_0\right) \\ &\text{s.t.} \quad \mathbf{Y}_0^T \mathbf{Y}_0 = \mathbf{I}_{K-1} \text{ and } \mathbf{Y}_0^T \boldsymbol{\Pi}^{1/2} \mathbf{1}_n = \mathbf{0}. \end{aligned} \tag{14}$$

The optimization problems (13)-(14) are similar to problems (4.2)-(4.3) in Zhang and Jordan (2008). The solution of problem (14) is given in the following theorem, which can be directly obtained from Appendix A.4 in Zhang and Jordan (2008).

Theorem 6. *Let the columns of* $\mathbf{U} \in \mathcal{R}^{n\times(K-1)}$ *be the first* $K-1$ *eigenvectors of* $-\Pi^{1/2}\mathbf{H}_\pi^T \boldsymbol{K}_P^\pi \mathbf{H}_\pi \mathbf{\Pi}^{1/2}$. *Then, the solution of problem* (14) *is* $\widehat{\mathbf{Y}}_0 = \mathbf{UQ}$, *where* $\mathbf{Q}$ *is an arbitrary* $(K-1)\times(K-1)$ *orthonormal matrix.*

It follows from Theorem 6 that the solution of problem (13) is $\widehat{\mathbf{Y}} = \mathbf{\Pi}^{-1/2}\mathbf{UQ}$. To recover the assignment matrix $\mathbf{E}$ in the problem (12), we can transform $\widehat{\mathbf{Y}}$ into a discrete set of values by the rounding scheme. Here, we use the K-means rounding algorithm proposed by Zhang and Jordan (2008). Then, we provide a spectral relaxation algorithm with K-means rounding for the optimization problem (12), listed in Algorithm 3.

Algorithm 3 Spectral relaxation with K-means rounding for the K-CDFs

1: **Relaxation:** Obtain $\mathbf{Y} = (\mathbf{y}_1,\ldots,\mathbf{y}_n)^T = \mathbf{\Pi}^{-1/2}\mathbf{UQ}$ from problem (13), where $\mathbf{Q} = \mathbf{I}_{K-1}$;

2: **Initialize:** Let $t = 0$. Choose the initial assignment matrix $\mathbf{E}^{(t)}$;

3: **Rounding:** Repeat the following procedure until convergence:

(a) Compute $\mathbf{m}_k^{(t)} = \frac{1}{\sum_{i\in V_k^{(t)}}\pi_i}\sum_{i\in V_k^{(t)}}\pi_i\mathbf{y}_i$, where $V_k^{(t)} = \{i \in \{1,2,\ldots,n\} : \mathbf{E}_{ik}^{(t)} = 1\}$;

(b) Find $k_i = \arg\min_{1\le j\le K-1}\left\|\mathbf{y}_i - \mathbf{m}_j^{(t)}\right\|$, and update $\mathbf{E}^{(t)}$ by allocating the i-th data point to class k_i. Set $t \leftarrow t+1$;

4: **Output:** Output $\mathbf{E}^{(t)}$.

We also use the Procrustean rounding scheme in Zhang and Jordan (2008). Let $\mathbf{G} = \left(\mathbf{I}_{K-1} - \frac{1}{K}\mathbf{1}_{K-1}\mathbf{1}_{K-1}^T, -\frac{1}{K}\mathbf{1}_{K-1}\right)^T$. Then, we can propose a spectral relaxation with Procrustean rounding listed in Algorithm 4. Through our limited numerical simulations, we find that it seems that Algorithm 3 is better than Algorithm 4; see Example 5 in the Monte Carlo simulations.

Algorithm 4 Spectral relaxation with Procrustean rounding for the K-CDFs

1: **Relaxation:** Calculate $\mathbf{U} \in \mathcal{R}^{n\times(K-1)}$ defined in Theorem 6;

2: **Initialize:** Let $t = 0$. Choose the initial assignment matrix $\mathbf{E}^{(t)}$;

3: **Rounding:** Repeat the following procedure until convergence:

(a) Implement the SVD of $\mathbf{U}^T\mathbf{E}^{(t)}\mathbf{G}$ as $\mathbf{U}^T\mathbf{E}^{(t)}\mathbf{G} = \mathbf{\Theta}^{(t)}\mathbf{\Lambda}^{(t)}\{\mathbf{V}^{(t)}\}^T$. Compute $\mathbf{Y}^{(t)} = \left(y_{ij}^{(t)}\right)_{n\times(K-1)} = \mathbf{\Pi}^{-1/2}\mathbf{U}\mathbf{Q}^{(t)}$, where $\mathbf{Q}^{(t)} = \mathbf{\Theta}^{(t)}\{\mathbf{V}^{(t)}\}^T$;

(b) Find $k_i^{(t)} = \arg\max_{1\le j\le K-1} y_{ij}^{(t)}$, and update $\mathbf{E}^{(t)}$ by allocating the i-th data point to class $k_i^{(t)}$ if $\max_{1\le j\le K-1} y_{ij}^{(t)} > 0$, and to class K otherwise. Set $t \leftarrow t+1$;

4: Output $\mathbf{E}^{(t)}$.

3.3 Related works

In this section, we first present some relationships between the K-CDFs and the energy statistic-based clustering (Li and Rizzo, 2017; Franca et al., 2021). By Lemma 6.1 in Kim et al. (2020), we define the angular distance between any $\mathbf{v}_1 \in \mathcal{R}^p$ and $\mathbf{v}_2 \in \mathcal{R}^p$ as

$$\rho_{\mathrm{W}}(\mathbf{v}_1, \mathbf{v}_2) = \mathbb{E}_{\mathbf{W}}\left\{\arccos\left(\frac{(\mathbf{v}_1-\mathbf{W})^T(\mathbf{v}_2-\mathbf{W})}{\|\mathbf{v}_2-\mathbf{W}\|_2\|\mathbf{v}_2-\mathbf{W}\|_2}\right)\right\}.$$

If the expectation is taken with respect to the Lebesgue measure ν, we obtain that

$$\rho_{\nu}(\mathbf{v}_1, \mathbf{v}_2) = \mathbb{E}_{\mathbf{W}\sim\nu}\left\{\arccos\left(\frac{(\mathbf{v}_1-\mathbf{W})^T(\mathbf{v}_2-\mathbf{W})}{\|\mathbf{v}_2-\mathbf{W}\|_2\|\mathbf{v}_2-\mathbf{W}\|_2}\right)\right\} = \gamma_d \|\mathbf{v}_1-\mathbf{v}_2\|_2, \tag{15}$$

where γ_d is a positive constant.

Using (15) and Theorem 5, we can formulate the optimization problem of the energy statistic-based clustering

$$\max_{\mathbf{E}}\left\{-\operatorname{tr}\left(\mathbf{E}^T\mathbf{\Pi}\mathbf{H}_\pi^T\boldsymbol{\rho}_{\text{energy}}\mathbf{H}_\pi\mathbf{\Pi}\mathbf{E}(\mathbf{E}^T\mathbf{\Pi}\mathbf{E})^{-1}\right)\right\}, \tag{16}$$

where the (i,j)-element of $\boldsymbol{\rho}_{\text{energy}} \in \mathcal{R}^{n\times n}$ is equal to $(\boldsymbol{\rho}_{\text{energy}})_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|_2$. In manners analogous to Algorithms 2-4, we can obtain the energy statistic-based Lloyd and weight spectral relaxation algorithms. Note that the above extensions are basically consistent with

the works in Li and Rizzo (2017) and Franca et al. (2021), except that our motivation is from the nonparametric ANOVA.

Lemma 6.1 in Kim et al. (2020) further suggests that $\rho_{\mathrm{W}}(\mathbf{v}_1, \mathbf{v}_2)$ is a negative type metric. Thus, the K-CDFs can be viewed as an extension of the energy statistic-based clustering in a new space of negative type (Sejdinovic et al., 2013; Franca et al., 2021). By our theoretical and experimental studies, the advantage of the extension is obvious; for example, the K-CDFs are not sensitive to the data dimensions, are robust to heavy-tailed data, and do not require moment conditions on data.

Replacing $-\mathbf{K}_{\mathrm{p}}^{\pi}$ in (12) by any symmetric positive definite kernel, problem (12) becomes a weighted kernel K-means (Zhang and Jordan, 2008), given by

$$\max_{\mathbf{E}} \left\{ \mathrm{tr}\left(\mathbf{E}^T \mathbf{\Pi} \mathbf{H}_{\pi}^T \mathbf{K} \mathbf{H}_{\pi} \mathbf{\Pi} \mathbf{E} (\mathbf{E}^T \mathbf{\Pi} \mathbf{E})^{-1} \right) \right\}, \tag{17}$$

where $\mathbf{K} \in \mathcal{R}^{n \times n}$ is the Gram matrix for a user-specified kernel function. However, it must be noted that kernel K-means depends heavily on the choice of the kernel and suffers from a lack of interpretability due to nonlinear mapping.

4 Monte Carlo simulations

In this section, we conduct Monte Carlo simulations to assess the finite sample performance of the CDF procedure implemented by Algorithm 2 (CDFs-Lloyd for short) and Algorithm 3 (CDFs-SR). We compare the two algorithms with other existing methods, including K-means++ (Kmeans), Gaussian mixture models (GMM), the energy statistic-based clustering in (16) (Energy-SR) and the kernel K-means in (17). For the kernel K-means, we focus on the two classical kernel functions: Gaussian kernel (Kernel-G-SR) and Laplace kernel (Kernel-Lap-SR), given by

$$k(\mathbf{x}_1, \mathbf{x}_2) = \exp(-\|\mathbf{x}_1 - \mathbf{x}_2\|^2/\gamma_1) \quad \text{and} \quad k(\mathbf{x}_1, \mathbf{x}_2) = \exp\left(-\|\mathbf{x}_1 - \mathbf{x}_2\|/\gamma_2\right),$$

where $\gamma_1, \gamma_2 > 0$ are tuning parameters.

To evaluate the performance of the algorithms, we use the adjusted Rand index (ARI, Hubert and Arabie (1985)) between the true cluster and the obtained partition from a specified algorithm. Another commonly used measurement is the normalized mutual information (NMI, Vinh et al. (2010)) in the literature. In the simulations, the performance of the NMI is consistent with that of the ARI; thus, we report only the results of the ARI. In each experiment, we report the mean of the ARIs over 500 replications with $K = 2$ and 5.

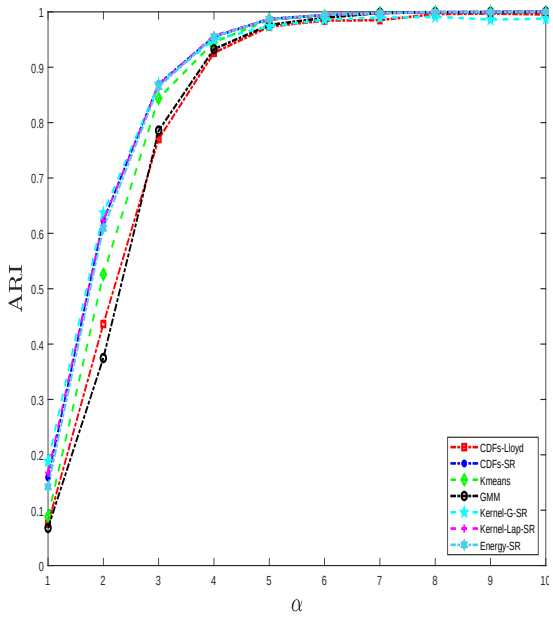
In the simulations, the weight vector is set to $\boldsymbol{\pi} = \mathbf{1}_n$. For the sake of fairness, all of the above algorithms are initialized with K-means++, implemented by calling the `K-means` function in MATLAB. The Energy-SR, Kernel-G-SR and Kernel-Lap-SR are implemented by the spectral relaxation algorithm with K-means rounding in Algorithm 3. Note that Gaussian and Laplace kernels depend on the parameters γ_1 and γ_2. We use the grid search to obtain the optimal parameters by maximizing their ARIs over the grids $\{0.01, 5, 10, 15, \ldots, 95, 100\}$.

Example 1. *In the example, we generate the data from the following multivariate normal mixtures:*

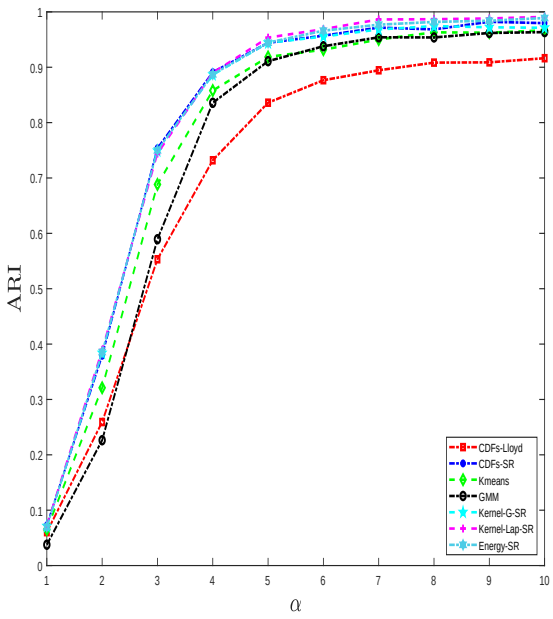
$$\mathbf{x}_1, \ldots, \mathbf{x}_n \overset{i.i.d.}{\sim} \frac{1}{K} N_p(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) + \cdots + \frac{1}{K} N_p(\boldsymbol{\mu}_K, \boldsymbol{\Sigma}_K),$$
$$\boldsymbol{\mu}_k = \alpha \mathbf{e}_{k,p}, \quad \boldsymbol{\Sigma}_k = \mathbf{I}_p, \quad k = 1, \cdots, K \quad (p \geq K),$$

where $N_p(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ is the p-dimensional normal distribution with the mean $\boldsymbol{\mu}_k$ and the covariance matrix $\boldsymbol{\Sigma}_k$ and $\alpha \in \mathcal{R}$ is a specified parameter.

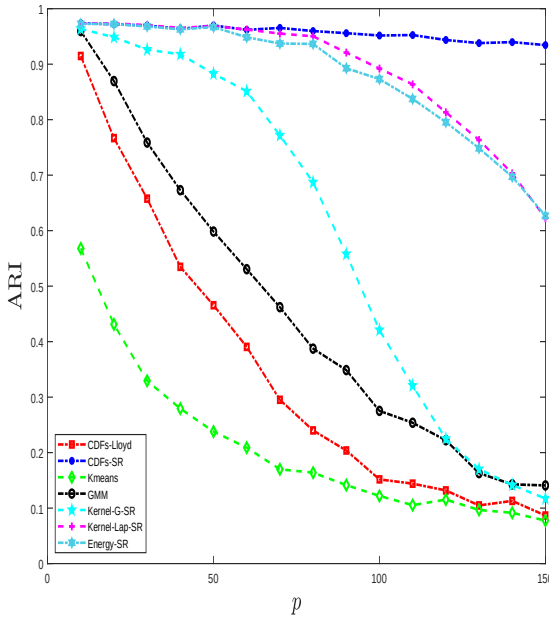
The average ARI over 500 replications comparisons are displayed in Figure 1. Figure 1(a)-(b) shows plots of the ARI curve against the location parameter α with $p = 5$. From Figure 1(a)-(b), we can see that the CDFs-SR, Energy-SR, Kernel-G-SR and Kernel-Lap-SR and Kmeans perform the best, followed by the GMM, and then the CDFs-Lloyd. The results suggest that the CDFs-Lloyd and CDFs-SR are comparable to the K-means on the Gaussian data.



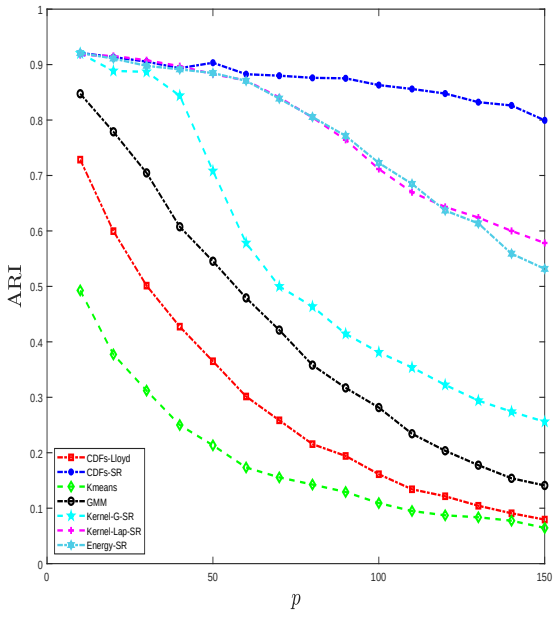
(a) $K = 2$, $p = 5$ and $n = 40$



(b) $K = 5$, $p = 5$ and $n = 100$

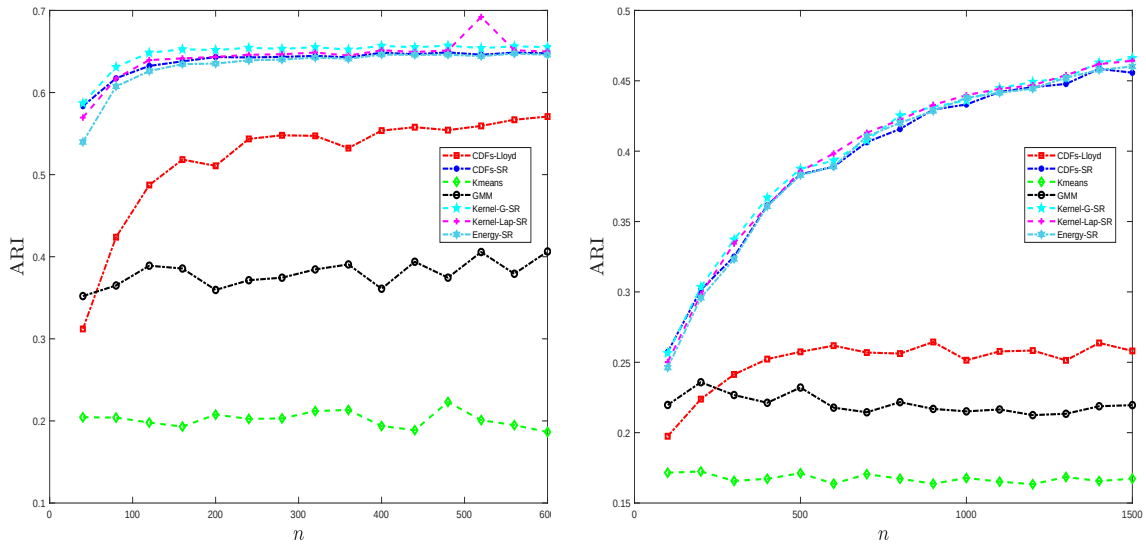


(c) $K = 2$, $\alpha = 4.5$ and $n = 40$



(d) $K = 5$, $\alpha = 4.5$ and $n = 100$

Figure 1: The average ARI comparisons for Example 1: (a)-(b): α varies; (c)-(d): p varies.



(a) $K = 2$, $\alpha = 2$ and $p = 5$ (b) $K = 5$, $\alpha = 2$ and $p = 5$

Figure 2: The average ARI comparisons with the sample size n for Example 1.

Figure 1(c)-(d) shows plots of the ARI curve against the dimension p with $\alpha = 4.5$. The results indicate that all methods degenerate as p increases. However, the CDFs-SR degrades much less and is more stable in high dimensions. We also provide plots of the ARI curve against the sample size n with $\alpha = 2$ and $p = 5$ in Figure 2. It can be seen that the CDFs-SR, Energy-SR, Kernel-G-SR and Kernel-Lap-SR have similar performances, and are better than the Kmeans and GMM. In addition, Figures 1-2 demonstrate that the CDFs-Lloyd are always inferior to the CDFs-SR, probably due to suffering from local minima.

Example 2. *The purpose of the example is to evaluate the finite sample performance of K-CDFs algorithms for heavy-tailed data. We generate the data from the multivariate t mixtures, given by*

$$\mathbf{x}_1, \ldots, \mathbf{x}_n \overset{i.i.d.}{\sim} \frac{1}{K}\boldsymbol{t}_{df}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) + \cdots + \frac{1}{K}\boldsymbol{t}_{df}(\boldsymbol{\mu}_K, \boldsymbol{\Sigma}_K),$$
$$\boldsymbol{\mu}_k = \alpha \mathbf{e}_{k,p}, \quad \boldsymbol{\Sigma}_k = \mathbf{I}_p, \quad k = 1, \cdots, K \quad (p \geq K),$$

where $\boldsymbol{t}_{df}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ is the p-dimensional t-distribution with the degree of freedom df. Here,

we consider df $= 3$ *and 1.*

The average ARI comparisons for Example 2 are summarized in Figure 3 for $df = 3$ and Figure 4 for $df = 1$. Figure 3(a)-(b) displays similar performance to Figure 1(a)-(b), except that Energy-SR is slightly inferior to CDFs-SR, Kernel-G-SR and Kernel-Lap-SR. Although the inferiority is not obvious, it is substantial owing to the heavy-tailed data. This phenomenon is clearly demonstrated in Figure 4(a)-(b) for the $\mathbf{t}_1$ mixtures. This may be because the condition $E[\|\mathbf{X}\|] < \infty$ required by the energy statistic is violated. Figure 4(a)-(b) also implies that the CDFs-SR and Kernel-Lap-SR are superior to the Kernel-G-SR. These results suggest that the CDFs-SR can automatically identify the characteristics of the underlying distribution.

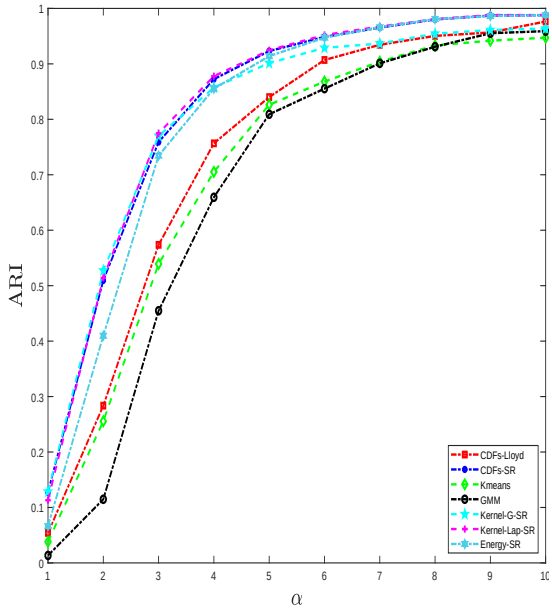
In Figure 3(c)-(d), we can see that the CDFs-SR degrades much less than the other methods as p increases. In Figure 4(c)-(d), the CDFs-SR outperforms the Energy-SR and is comparable to the Kernel-G-SR and Kernel-Lap-SR when p is not too large. It is noteworthy that although Kernel-G-SR and Kernel-Lap-SR have high performance, they depend heavily on the choice of γ_1 and γ_2, which poses a significant computational burden.

Example 3. *In the example, we evaluate the finite sample performance of* K*-CDFs algorithms for skewed data. The datasets are generated by the following multivariate lognormal mixtures:*

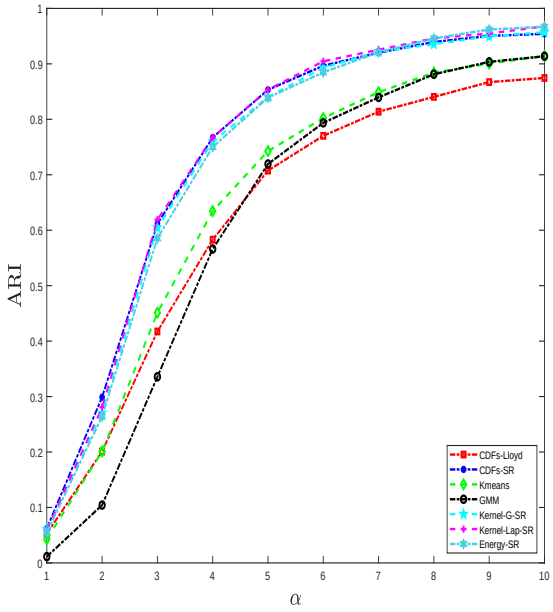
$$\mathbf{x}_1, \ldots, \mathbf{x}_n \overset{i.i.d.}{\sim} \frac{1}{K} e^{N_p(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)} + \cdots + \frac{1}{K} e^{N_p(\boldsymbol{\mu}_K, \boldsymbol{\Sigma}_K)},$$

where $\boldsymbol{\mu}_k = \alpha \mathbf{e}_{k,p}$ *and* $\boldsymbol{\Sigma}_k = \mathbf{I}_p$*, for* $\alpha \in \mathcal{R}$*,* $k \in \{1, \cdots, K\}$ *and* $K \leq p$*.*

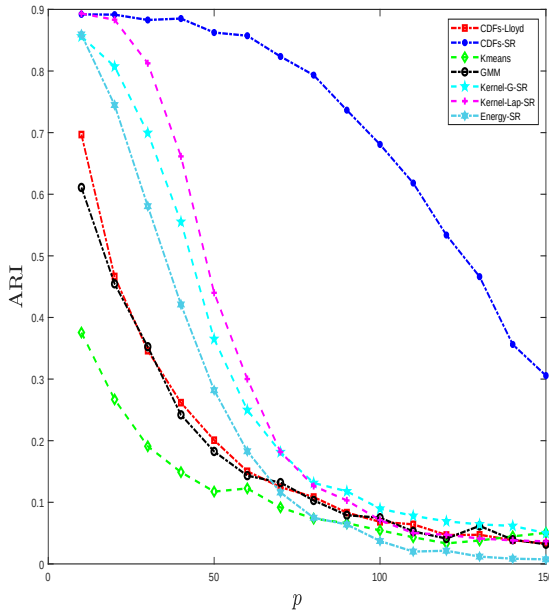
In Figure 5, it can be seen that the CDFs-SR and even CDFs-Lloyd perform noticeably better than the other five methods. The results indicate that the two K-CDFs algorithms work well when the datasets are strongly skewed in the settings. Somewhat surprisingly, in Figure 5(a)-(b), the ARI curves for the Kernel-G-SR and Kernel-Lap-SR change nonmonotonically with the parameter α and have inferior performance to the GMM for $\alpha \geq 5$.



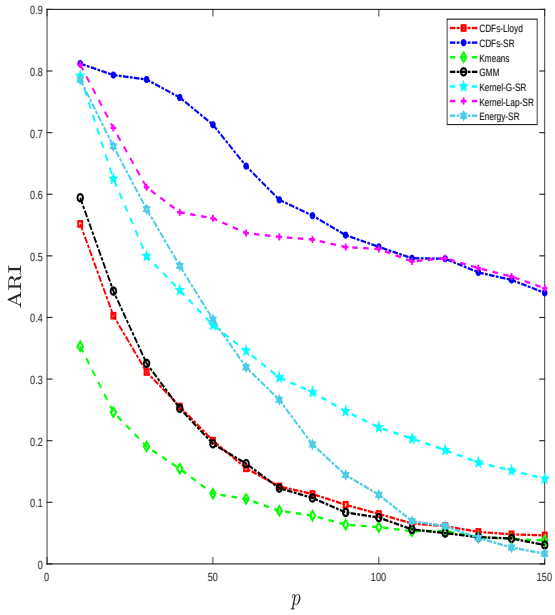
(a) $K = 2$, $p = 5$ and $n = 40$



(b) $K = 5$, $p = 5$ and $n = 100$

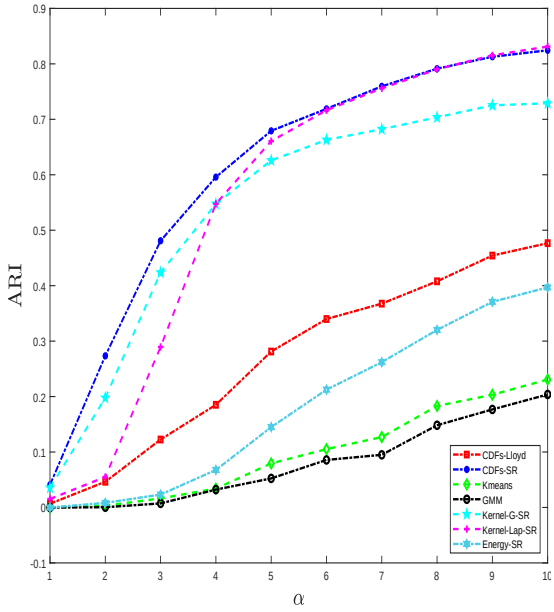


(c) $K = 2$, $\alpha = 4.5$ and $n = 40$

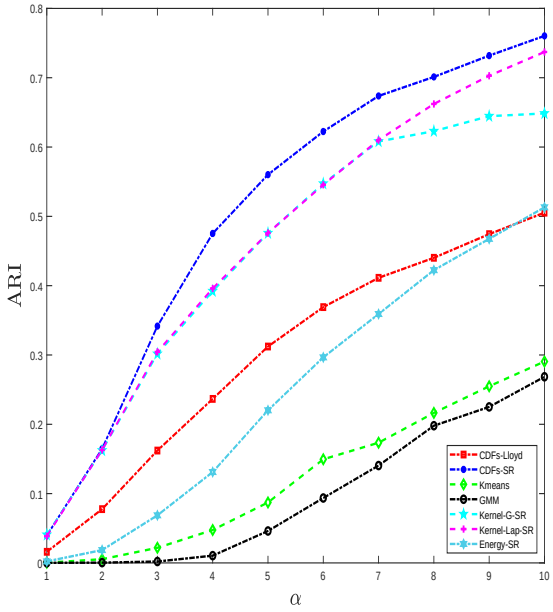


(d) $K = 5$, $\alpha = 4.5$ and $n = 100$

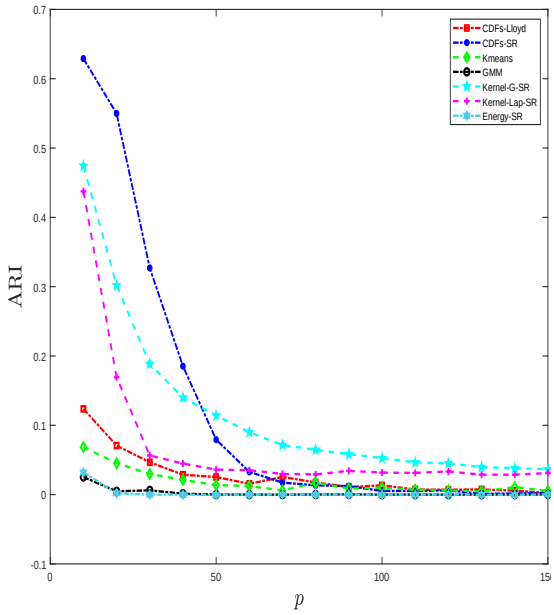
Figure 3: The average ARI comparisons for Example 2 with the $\mathbf{t}_3$ mixtures, i.e., $df = 3$: (a)-(b): α varies; (c)-(d): p varies.



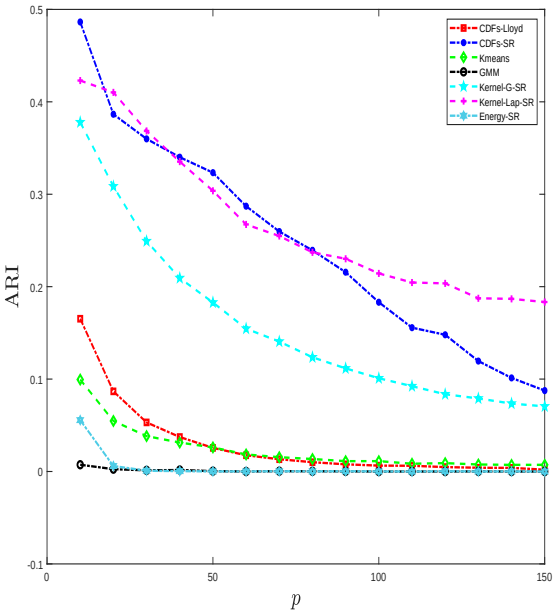
(a) $K = 2$, $p = 5$ and $n = 40$



(b) $K = 5$, $p = 5$ and $n = 100$

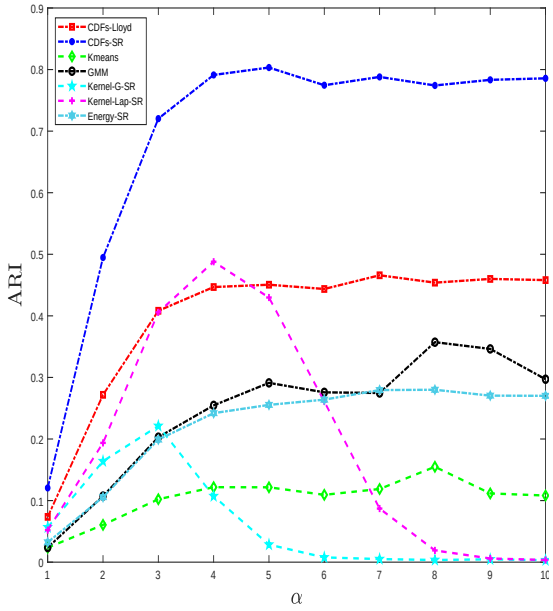


(c) $K = 2$, $\alpha = 4.5$ and $n = 40$

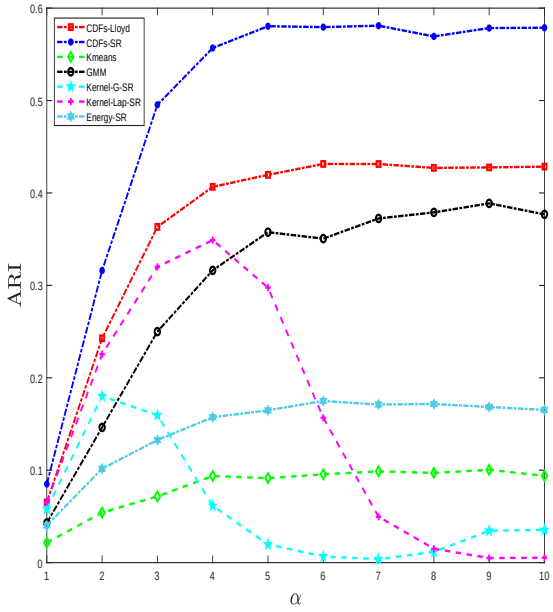


(d) $K = 5$, $\alpha = 4.5$ and $n = 100$

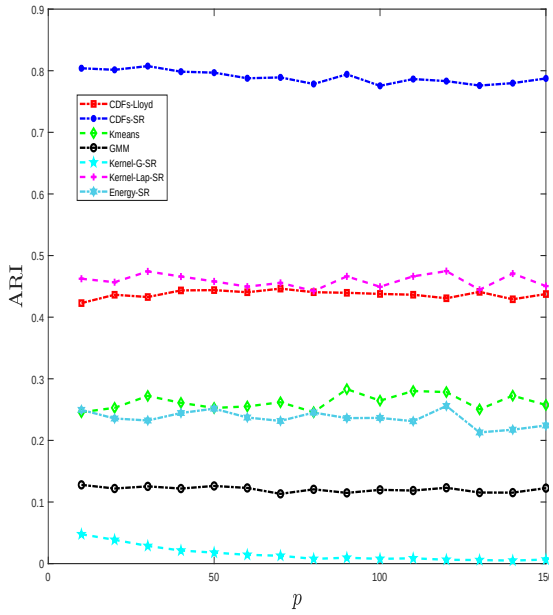
Figure 4: The average ARI comparisons for Example 2 with the $\mathbf{t}_1$ mixtures, i.e., $df = 1$: (a)-(b): α varies; (c)-(d): p varies.



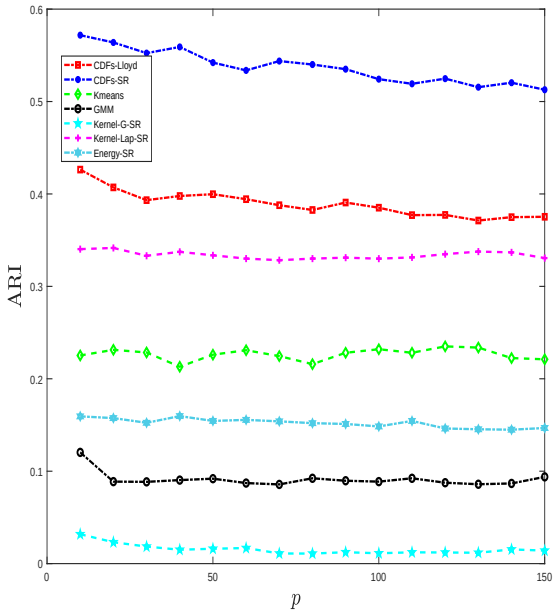
(a) $K = 2$, $p = 5$ and $n = 40$



(b) $K = 5$, $p = 5$ and $n = 100$



(c) $K = 2$, $\alpha = 4.5$ and $n = 40$



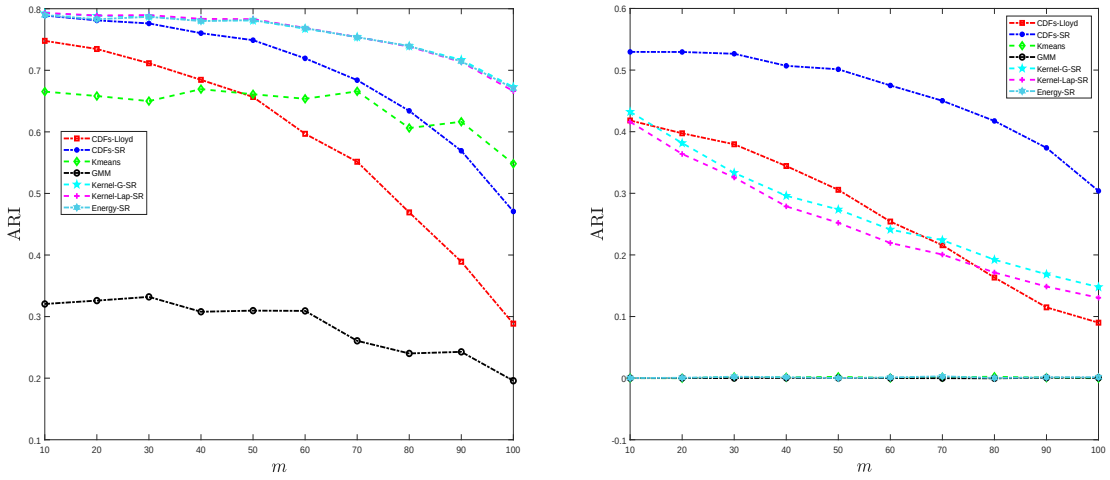
(d) $K = 5$, $\alpha = 4.5$ and $n = 100$

Figure 5: The average ARI comparisons for Example 3: (a)-(b): α varies; (c)-(d): p varies.

Example 4. *In the example, we generate the data from the multivariate t mixtures with unbalanced clusters, given by*

$$\mathbf{x}_1, \ldots, \mathbf{x}_n \overset{i.i.d.}{\sim} \frac{N-m}{2N} \boldsymbol{t}_{df}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) + \frac{N+m}{2N} \boldsymbol{t}_{df}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2), \quad N = 150,$$
$$\boldsymbol{\mu}_1 = (0,0,0,0,0)^T, \quad \boldsymbol{\mu}_2 = (2,2,2,2,2)^T, \quad \boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \mathbf{I}_5,$$

where m is a positive integer to control the unbalanced proportion of the two clusters.



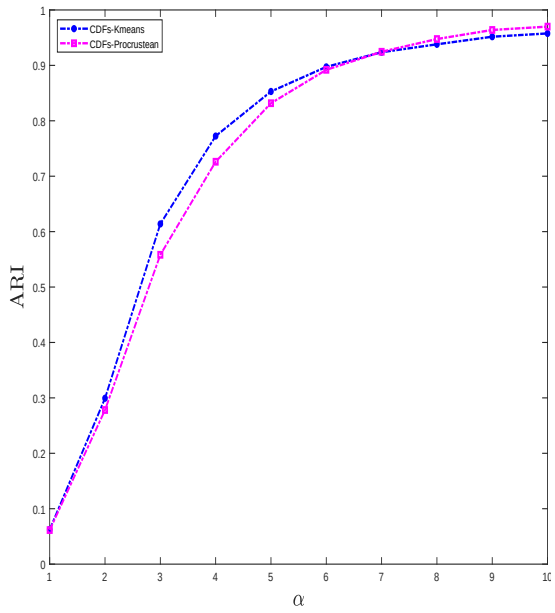
(a) $df = 3$ and $p = 5$ (b) $df = 1$ and $p = 5$

Figure 6: The average ARI comparisons with m for Example 4.

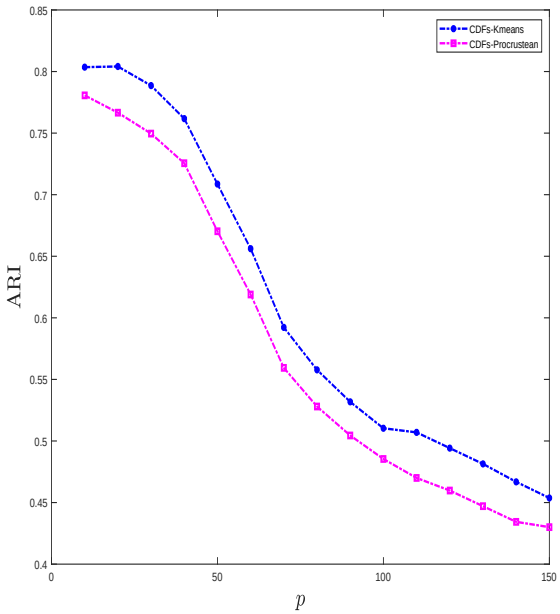
Figure 6 shows plots of the average ARI against $m \in [10, 100]$ for $df = 1$ and 3. Note that clusters become more unbalanced as m increases. It can be seen in Figure 6 that for highly unbalanced clusters, the Kernel-G-SR, Kernel-Lap-SR and Energy-SR are better than the CDFs-SR for $df = 3$, whereas the CDFs-SR performs the best for $df = 1$.

Example 5. *In the example, we compare Algorithm 3 with Algorithm 4. The datasets are generated from Example 2.*

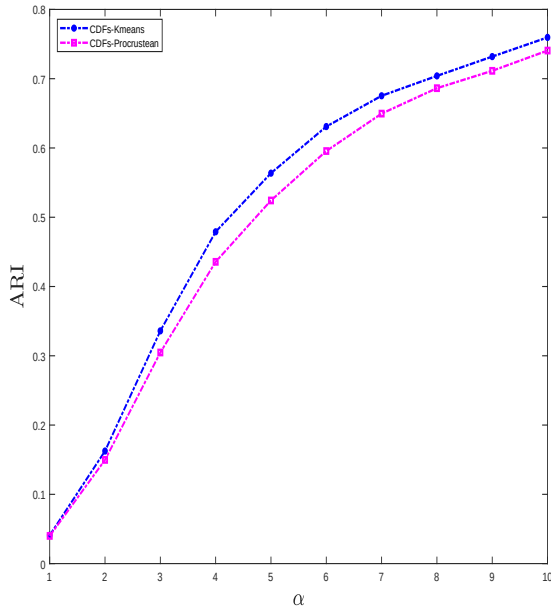
In Figure 7, Algorithms 3 and 4 are denoted by CDFs-kmeans and CDFs-Procrustean, respectively. In Figure 7, it can be seen that Algorithm 3 outperforms Algorithm 4 in most cases. The performance for other values of K is similar, and thus, we report only the



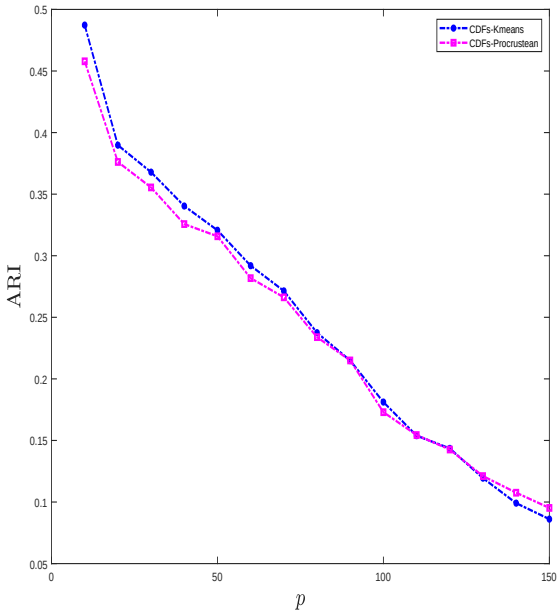
(a) $df = 3$, $p = 5$ and $K = 5$



(b) $df = 3$, $\alpha = 4.5$ and $K = 5$



(c) $df = 1$, $p = 5$ and $K = 5$



(d) $df = 1$, $\alpha = 4.5$ and $K = 5$

Figure 7: The average ARI comparisons with $n = 100$ for Example 5.

results for $K = 5$. In addition, some extra experiments show that CDFs-kmeans is faster than CDFs-Procrustean (results not reported here).

From the above numerical results, we obtain the following findings. (1) As expected, the CDFs-SR outperforms the CDFs-Lloyd in all settings, which suggests that the spectral relaxation method can effectively avoid local minima; (2) When the data are generated from heavy-tailed distributions with infinite moments, the CDFs-SR and even CDFs-Lloyd are superior to the Energy-SR, mainly because the assumption of $E[\|\mathbf{X}\|] < \infty$ required by the Energy-SR is not satisfied; (3) Although the kernel K-means is comparable to the K-CDF algorithms in some settings, it depends heavily on the choice of the kernel function and tuning parameter. In contrast, the K-CDF algorithms are free of tuning parameters and can automatically capture the data characteristics.

5 Real data analysis

In this section, we illustrate the K-CDF algorithms by empirical analysis of several real datasets. We first consider Dermatology data, which include 366 observations and 34 attributes. Deleting 8 observations with missing entries in the age column, we obtain 358 complete observations. Note that the data have 6 classes. According to the prior information, we determine the number of clusters $K = 6$. That is, $(n, p, K) = (358, 34, 6)$. The description of the data can be found in the UCI Machine Learning Repository.

Similar to previous simulation studies, each algorithm is initialized with K-means++. Here, we perform each algorithm over 100 replications by the different initial values. Table 1 reports the averages of the 100 ARIs and 100 NMIs by each algorithm for Dermatology data. In Table 1, we can see that the CDFs-SR outperforms the other methods by higher ARI and NMI values. In addition, we further present heatmaps of the cluster membership by the CDFs-Lloyd and CDFs-SR in Figure 8, which indicates that the CDFs-SR exhibits better performance for Dermatology data.

Table 1: The averages of the ARI and NMI over 100 replications for Dermatology data

	CDFs-Lloyd	CDFs-SR	Kmeans	GMM	Kernel-G-SR	Kernel-Lap-SR	Energy-SR
ARI	0.8191	**0.9129**	0.6950	0.7870	0.8049	0.8190	0.8569
NMI	0.8903	**0.9150**	0.7608	0.8136	0.8473	0.8527	0.8801

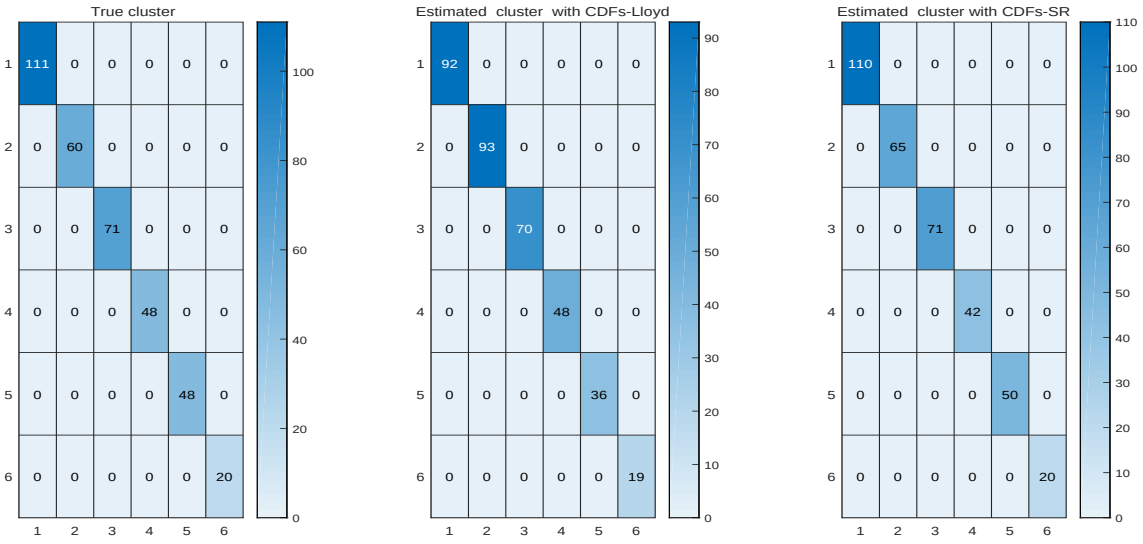


Figure 8: Heatmaps of the cluster estimated by the CDFs-Lloyd and CDFs-SR for Dermatology data.

We next consider the following seven datasets from the UCI Repository: 1. Wine Dataset—(Wine); 2. Statlog (Heart) Dataset—(Heart); 3. Libras Movement Dataset—(Libras); 4. Diagnostic Wisconsin Breast Cancer—(Wbdc); 5. Breast Cancer Wisconsin (Original) Dataset—(Wisconsin); 6. Breast Tissue Dataset—(Breast); 7. Seeds Dataset—(Seeds). Further details can be found at the website: https://archive.ics.uci.edu/ml/datasets.php.

Table 2 reports the average ARI and NMI over 100 replications for the seven real datasets, where all K's are determined by the prior information on the class of the data. The results suggest that the CDFs-SR and CDFs-Lloyd methods outperform the other methods in most cases. To further illustrate the stability of the algorithms, we provide boxplots of the 100 ARIs for each real dataset in Figure 9. We can see that the CDFs-SR and Energy-SR enjoy smaller variations in all cases, whereas the Kernel-G-SR and Kernel-Lap-SR have large variations. This may be because the kernel methods need to choose the turning parameters in each replication, which may cause large variation.

6 Discussion

We propose a new CDF-based clustering procedure, K-CDFs, which generalizes the classical K-means and represents the cluster centers by empirical CDFs. It is natural to develop a Lloyd-type algorithm to implement the K-CDFs. We establish the convergence of the Lloyd-type algorithm. To avoid the local minima of the Lloyd-type algorithm, we propose the spectral relaxation algorithm by reformulating the K-CDFs as a trace maximization problem. Numerical results show the usefulness of the K-CDFs, which can outperform existing counterparts.

To conclude, we reiterate several nice properties of the K-CDFs. First, it is convenient to implement with no tuning parameters. Algorithms 2, 3 and 4 are simple and straightforward, similar to classical Lloyd's algorithm for K-means and the spectral re-

Table 2: The averages of the ARI and NMI over 100 replications for seven real datasets

Datasets $(n/p/K)$		CDFs-Lloyd	CDFs-SR	Kmeans	GMM	Kernel-G-SR	Kernel-Lap-SR	Energy-SR
Wine	ARI	**0.9143**	0.8828	0.6126	0.9002	0.7249	0.7480	0.8930
(178/13/3)	NMI	**0.7133**	0.6994	0.5290	0.7056	0.5808	0.6035	0.6965
Heart	ARI	0.3537	**0.4538**	0.2293	0.3855	0.2957	0.2897	0.4322
(270/13/2)	NMI	0.2834	**0.3787**	0.1982	0.3008	0.2801	0.2767	0.3575
Libras	ARI	**0.3143**	0.3074	0.2975	0.2720	0.2143	0.2718	0.2966
(360/90/15)	NMI	**0.5872**	0.5828	0.5629	0.5457	0.5170	0.5536	0.5749
Wbdc	ARI	0.6140	0.7022	0.0853	**0.8185**	0.6099	0.5511	0.6779
(569/30/2)	NMI	0.4262	0.4297	0.0632	**0.5182**	0.3626	0.3269	0.4292
Wisconsin	ARI	0.9061	**0.9081**	0.8419	0.5767	0.8762	0.8497	0.8914
(683/9/2)	NMI	0.8449	**0.8467**	0.7489	0.5668	0.7906	0.7592	0.8169
Breast	ARI	0.3456	**0.4072**	0.2363	0.3816	0.2667	0.1875	0.3211
(106/9/6)	NMI	**0.5743**	0.5522	0.4967	0.5712	0.4773	0.4091	0.5053
Seeds	ARI	**0.8239**	0.7478	0.7767	0.8137	0.6949	0.6868	0.7974
(210/7/3)	NMI	**0.7865**	0.7084	0.7441	0.7629	0.6992	0.6890	0.7591

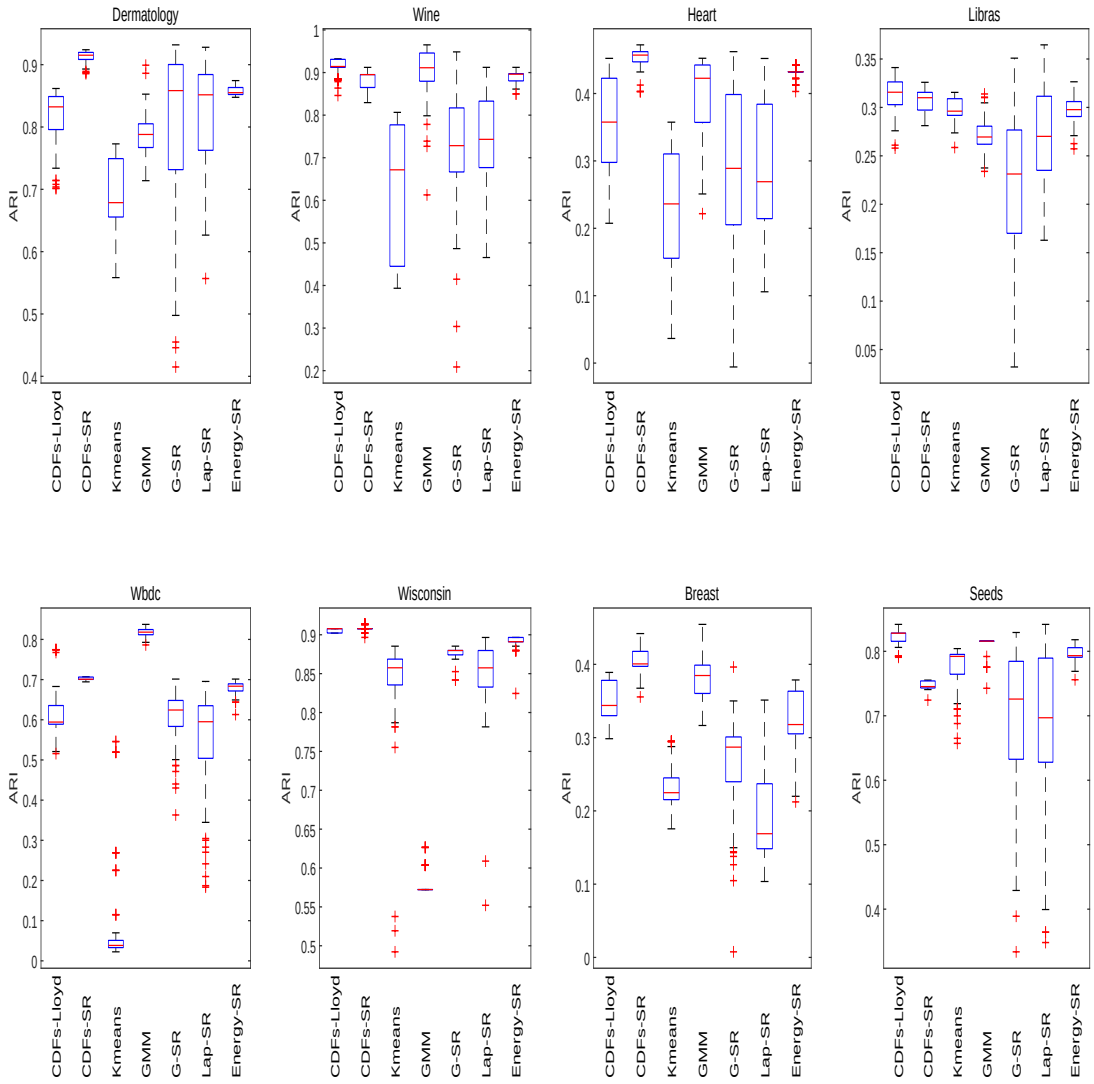


Figure 9: Boxplots of the ARIs over 100 replications obtained for seven real datasets.

laxation algorithm (Zhang and Jordan, 2008). Second, the K-CDFs are nonparametric and model-free, unlike K-means and GMM, which depend on the assumptions on cluster distributions. Moreover, the K-CDFs can separate clusters when data are nonlinearly separable. Third, the K-CDFs are robust to heavy-tailed or skewed data. The K-CDFs can automatically detect the characteristics of data and exhibit good cluster recovery, whereas the kernel K-means needs to choose the proper kernel function and tuning parameter to capture the characteristics of data. Finally, the K-CDFs method has a good statistical interpretation, such as K-means. In summary, the K-CDFs enjoy all the appealing features confirmed by Zhu et al. (2017), Kim et al. (2020) and Liu et al. (2022) can be viewed as their generalization to the unsupervised learning problem.

Although we assume that the data are continuous in the above sections, Algorithms 1 and 2 can be extended to discrete data. However, for discrete data, it can be seen that $P(\beta^T \mathbf{x}_1 \leq \beta^T \mathbf{x}_2) > 0$ for any $\beta \in \mathbb{S}^{p-1}$ due to the existence of ties, which is different from that for continuous data. Thus, for discrete data, the matrix $\mathbf{K}_\mathbf{P}$ in Theorem 1 needs to be improved. We can use the arguments given in the proof of Lemma B.1 in Kim et al. (2020) to handle this problem, which can provide a general result applicable both for continuous and discrete data. The detailed analysis is left to future research.

Supplementary material

The online Supplementary Material contains proofs of the theoretical results and the code in MATLAB to replicate all numerical experiments.

Acknowledgement

The authors thank the associate editor and two anonymous reviewers for provided helpful comments on earlier drafts of the manuscript. Jicai Liu's research was supported by

the National Science Foundation of China (11426156, 11501372, 11971018) and the Open Research Fund of Key Laboratory of Advanced Theory and Application in Statistics and Data Science (East China Normal University), Ministry of Education (KLATASDS2001). Riquan Zhang's research was supported by the National Natural Science Foundation of China (11971171, 11571112).

Disclosure statement

The authors report there are no competing interests to declare.

References

Chan, P. K., Schlag, M. D. F., and Zien, J. Y. (1994). Spectral k-way ratio-cut partitioning and clustering. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 13(9):1088–1096.

Dhillon, I. S., Guan, Y., and Kulis, B. (2004). Kernel k-means: spectral clustering and normalized cuts. *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 551–556.

Escanciano, J. C. (2006). A consistent diagnostic test for regression models using projections. *Econometric Theory*, 22(6):1030–1051.

Filippone, M., Camastra, F., Masulli, F., and Rovetta, S. (2008). A survey of kernel and spectral methods for clustering. *Pattern Recognition*, 41(1):176–190.

Fraley, C. and Raftery, A. E. (2002). Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association*, 97(458):611–631.

Franca, G., Rizzo, M., and Vogelstein, J. T. (2021). Kernel k-groups via Hartigan's method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(12):4411–4425.

Hubert, L. and Arabie, P. (1985). Comparing partitions. *Journal of Classification*, 2:193–218.

Kim, I., Balakrishnan, S., and Wasserman, L. (2020). Robust multivariate nonparametric tests via projection averaging. *Annals of Statistics*, 48(6):3417–3441.

Li, S. and Rizzo, M. L. (2017). K-groups: a generalization of k-means clustering. *arXiv:1711.04359*.

Liu, J., Si, Y., Xu, W., and Zhang, R. (2022). A new nonparametric extension of ANOVA via projection mean variance measure. *Statistica Sinica*, 32(1):367–390.

Lloyd, S. (1982). Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137.

MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 1:281–297.

Pelleg, D. and Moore, A. W. (2000). X-means: extending k-means with efficient estimation of the number of clusters. *Proceedings of the Seventeenth International Conference on Machine Learning*, 1:727–734.

Pollard, D. (1981). Strong consistency of k-means clustering. *Annals of Statistics*, 9(1):135 – 140.

Schölkopf, B., Smola, A., and Müller, K. R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319.

Sejdinovic, D., Sriperumbudur, B. K., Gretton, A., and Fukumizu, K. (2013). Equivalence of distance-based and RKHS-based statistics in hypothesis testing. *Annals of Statistics*, 41(5):2263–2291.

Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905.

Steinhaus, H. (1956). Sur la division des corps matériels en parties. *Bulletin L'Académie Polonaise des Science*, 4:801–804.

Thorndike, R. (1953). Who belongs in the family? *Psychometrika*, 18(4):267–276.

Tibshirani, R., Walther, G., and Hastie, T. (2001). 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.

Vinh, N. X., Epps, J., and Bailey, J. (2010). Information theoretic measures for clusterings comparison: variants, properties, normalization and correction for chance. *Journal of Machine Learning Research*, 11(95):2837–2854.

Zha, H., He, X., Ding, C., Simon, H., and Ming, G. (2002). Spectral relaxation for k-means clustering. *Advances in neural information processing systems*, 14:1057–1064.

Zhang, Z. and Jordan, M. I. (2008). Multiway spectral clustering: a margin-based perspective. *Statistical Science*, 23(3):383–403.

Zhu, L., Xu, K., Li, R., and Zhong, W. (2017). Projection correlation between two random vectors. *Biometrika*, 104(4):829–843.