Statistica Sinica Preprint No: SS-2025-0153	
Title	Spectral Clustering for Functional Data with Asymptotic Properties
Manuscript ID	SS-2025-0153
URL	http://www.stat.sinica.edu.tw/statistica/
DOI	10.5705/ss.202025.0153
Complete List of Authors	Yang Ren, Jinguan Lin, Peijun Sang and Jiangyan Wang
Corresponding Authors	Jiangyan Wang
E-mails	wangjiangyan2007@126.com
Notice: Accepted author version.	

Spectral Clustering for Functional Data with Asymptotic Properties

Yang Ren[1], Jinguan Lin[1], Peijun Sang[2] and Jiangyan Wang[1,*]

1. Nanjing Audit University; 2. University of Waterloo

*Corresponding author: wangjiangyan2007@126.com

Abstract:

Traditional clustering methods are mainly developed for multivariate data and often struggle to capture the continuity and dynamics of functional data, thus neglecting key features. To address this issue, we propose a two-stage functional data spectral clustering (TSFDSC) method that can effectively cluster both dense and sparse functional data using the graph Laplacian operator. For dense functional data, they are projected onto a set of pre-determined basis functions, and then the resulting coefficients are clustered by spectral clustering, whereas for sparse functional data, functional principal component scores are used instead. By leveraging low-dimensional representations from spectral embedding, TSFDSC achieves computational efficiency. We establish asymptotic properties for the proposed procedure. Simulation studies demonstrate that TSFDSC outperforms some commonly used functional clustering methods under various settings. The effectiveness of TSFDSC is further illustrated by two real applications.

Key words and phrases: Spectral clustering; graph Laplacian; functional data analysis; similarity matrix.

1. Introduction

Clustering is a fundamental technique in data mining, employed as an unsupervised learning method to partition datasets into meaningful subgroups. It finds broad applications across disciplines such as statistics, computer science and psychology. Functional data, represented as curves, images or other multidimensional functions (Li and Yang,

2023; Liang et al., 2023; Ma et al., 2024), are often assumed to reside in an infinite-dimensional space. This poses both theoretical and computational challenges. Extensive research has been conducted in machine learning with functional data; see Wang et al. (2016) for a detailed review. In particular, the identification of homogeneous subgroups of functional data via clustering or classification has garnered significant attention (Huang and Ruppert, 2023; Xue et al., 2024). It should be noted that traditional multivariate clustering methods often fail to account for the continuity and dynamics inherent in functional data, leading to poor performance. In contrast, functional clustering methods that can exploit the unique features of functional data offer a promising alternative.

Despite early advances in functional data analysis (FDA), progress in functional clustering analysis has been relatively slow since the 1980s (Zhang and Parnell, 2023). Functional clustering methods can be classified into four main types: raw data methods, filtering methods, adaptive methods, and distance-based methods (Jacques and Preda, 2014). Raw data methods apply traditional multivariate clustering techniques such as k-means and hierarchical clustering to the discrete observations made in each curve. Obviously such methods do not take the continuity and dynamics inherent in functional data into consideration. Moreover, the performance of such methods is sensitive to the choice of initial conditions, and they are not applicable to nonconvex shapes. Last but not least, they treat the curves as vectors, whereas the observation times (locations) can vary from curve to curve, which poses challenges to these methods. In contrast, filtering methods first represent functional data in terms of a set of basis functions, and then traditional multivariate clustering methods are employed to cluster the basis expansion coefficients. For example, Abraham et al. (2003) considered the B-spline basis functions for the representation of curves and then applied k-means clustering to the coefficients. In adaptive methods, these basis expansion coefficients are treated as random vectors following a mix-

ture model. For instance, James and Sugar (2003) clustered curves based on a Gaussian mixture model for the coefficients. Chiou and Li (2007) proposed an adaptive method called k-centres functional clustering (kCFC). Instead of using a conditional probability, kCFC uses the L^2 distance between the curves and thus determines the relative likelihood of the given curve belonging to different clusters. Furthermore, based on the idea of kCFC, Chiou and Li (2008) proposed a correlation-based clustering method to group curves with similar shapes. These methods usually suffer from high computational complexity, since they simultaneously perform dimensionality reduction through the basis expansion and clustering by fitting the mixture model. For distance-based methods, it can be related to either raw data or filtering methods, based on the different distance calculation methods. Ieva et al. (2013) applied k-means clustering to functional data directly based on the distance between curves and the distance between the derivative of the curves. But it should be noted that they are highly vulnerable to noise contained in the observations.

Consider a scenario where functional data are observed at a high sampling frequency, yielding dense functional data. Functional clustering is particularly effective in this context due to the richness of the observations. However, challenges arise with sparse functional data, as traditional nonparametric smoothing techniques often fail to provide a good representation for the underlying true curve from sparse observations. To address these challenges, James and Sugar (2003) proposed a flexible model-based clustering approach for functional data. Despite these innovations, model-based approaches still encounter difficulties due to data sparsity, which makes parameter estimation in models challenging. Given these limitations, it is desirable to develop a clustering method that is applicable to both dense and sparse functional data while maintaining accuracy and interpretability. To this end, spectral clustering (Von Luxburg, 2007) emerges as a promising alternative. For multivariate data, spectral clustering leverages graph structures and performs eigen-

decomposition on the graph Laplacian for dimension reduction and clustering. To our best knowledge, few studies have considered spectral clustering in functional data clustering, despite its widespread use in high-dimensional data analysis. Al Alawi (2021) introduced spectral clustering for functional data based on distance measures, though this approach faces challenges in calculating distances.

In this article, our objective is to develop a clustering method that can handle both dense and sparse functional data. To this end, we propose two-stage functional data spectral clustering (TSFDSC), a functional spectral clustering algorithm based on the graph Laplacian. For densely observed functional data, TSFDSC first performs basis expansion for functional data, representing the data in terms of a common set of pre-determined basis functions, and then clusters the resulting basis coefficients using spectral clustering. Grounded in the spectral graph theory, TSFDSC efficiently clusters data in arbitrarily shaped sample spaces. One potential concern is that the basis representation step via non-parametric smoothing is not reliable for sparse functional data in TSFDSC. To address this concern, we adopt functional principal component analysis (FPCA) proposed by Yao et al. (2005), and then use estimated FPC scores as the coefficients. These coefficients will be used to construct the similarity matrix and perform spectral clustering. For TSFDSC, we establish asymptotic properties for densely observed functional data. Moreover, we conduct a comparative and sensitivity analysis between functional data k-means (FD k-means) and TSFDSC for dense, regular sparse and irregular sparse functional data. These numerical studies demonstrate that, compared to k-means, our proposed method offers superior adaptability to diverse data distributions and generally produces better clustering results, demonstrating that TSFDSC is suitable for both sparse and dense functional data. The main contributions of this paper are two fold. First, it integrates functional data with spectral clustering within the framework of filtering methods. The simula-

tion results demonstrate that TSFDSC performs effectively on both dense and sparse functional data. Second, leveraging the convergence of the estimated spline coefficient vector, we establish the consistency of the estimated loss function, further validating the effectiveness of TSFDSC for dense functional data.

The remainder of this paper is structured as follows. Section 2 introduces the proposed TSFDSC based on B-spline smoothing or FPC scores, and presents the main theoretical results of the loss function. Section 3 conducts extensive simulations to illustrate the advantages of the proposed methods. In Section 4, we apply the proposed method to the Berkeley Growth Study data and CD4 data. This paper ends with some conclusions and discussion in Section 5. The technical proofs and some additional numerical results are deferred to the supplementary file.

2. Main results

In this section, we first present the functional data spectral clustering method and then establish its theoretical properties. To start, we introduce the basic framework for modeling functional data.

Let $\{\eta(x), x \in \chi\}$ be a stochastic process defined on a compact interval χ satisfying $\mathrm{E}\left\{\int_\chi \eta^2(x)dx\right\} < \infty$. Then $\eta(\cdot)$ can be treated as a mapping from a probability space $(\Omega, \mathscr{A}, P)$ to $(L^2(\chi), \mathscr{B})$, where $L^2(\chi)$ denotes the collection of all square-integrable functions defined on χ and $\mathscr{B}$ is the Borel σ-algebra. Consider a collection of n trajectories $\{\eta_i(x)\}_{i=1}^n$, which are i.i.d realizations of $\eta(x)$, with mean and covariance functions, $\mu(x) = \mathrm{E}\{\eta(x)\}$ and $G(x, x') = \mathrm{Cov}\{\eta(x), \eta(x')\}$, respectively. Each trajectory can be decomposed as $\eta_i(x) = \mu(x) + Z_i(x)$, where $Z_i(x)$ denotes the variation of the i-th trajectory from μ at x. Thus, $\mathrm{E}Z_i(x) = 0$ and $G(x, x') = \mathrm{Cov}\{Z_i(x), Z_i(x')\}$ for any $x, x' \in \chi$.

Under mild conditions, there exist non-increasing eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots > 0$

and corresponding eigenfunctions $\{\psi_k\}_{k=1}^{\infty}$, which constitute an orthonormal basis of $L^2(\chi)$, such that $G(x,x') = \sum_{k=1}^{\infty} \lambda_k \psi_k(x)\psi_k(x')$. Letting $\phi_k(x) = \sqrt{\lambda_k}\psi_k(x)$ for $k \geq 1$, $\eta(x)$ then admits the well-known Karhunen-Loève representation: $\eta(x) = \mu(x) + \sum_{k=1}^{\infty} \xi_k \phi_k(x)$, in which ξ_k, known as FPC scores, satisfy $E(\xi_k) = 0$ and $\mathrm{Cov}(\xi_k, \xi_j) = 1$ if $k = j$ and 0 otherwise. The rescaled eigenfunctions $\phi_k(x)$, known as FPCs, satisfy $\int G(x,x')\phi_k(x')dx' = \lambda_k \phi_k(x)$ for $k \geq 1$.

In reality, observations of functional data are usually contaminated with random noises. Let $\{(Y_{ij}, X_{ij}),\ 1 \leq i \leq n,\ 1 \leq j \leq m_i\}$ be observations made on a random sample of n experimental units, where Y_{ij} is the response observed on the i-th unit at X_{ij}. The observed data are then modeled as

$$Y_{ij} = \eta_i(X_{ij}) + \epsilon_{ij} = \mu(X_{ij}) + Z_i(X_{ij}) + \epsilon_{ij} = \mu(X_{ij}) + \sum_{k=1}^{\infty} \xi_{ik}\phi_k(X_{ij}) + \epsilon_{ij}, \qquad (2.1)$$

where ϵ_{ij}, $i = 1, \ldots, n$, $j = 1, \ldots, m_i$ are i.i.d random errors with mean 0 and variance σ_ϵ^2, and are independent of $Z_i(\cdot)$'s. In this article, for densely observed functional data, without loss of generality, $\eta_i(\cdot)$ is assumed to be recorded on a regular grid in $\chi = [0,1]$, and $X_{ij} = j/m_i$, $1 \leq j \leq m_i$ with m_i diverging to infinity as $n \to \infty$. This type of functional data was considered in Cao et al. (2016), Wang et al. (2020) and Wang et al. (2022), among others. Consequently, our observed data can be written as

$$Y_{ij} = \mu(j/m_i) + \sum_{k=1}^{\infty} \xi_{ik}\phi_k(j/m_i) + \epsilon_{ij}, \quad i = 1, \ldots, n,\ j = 1, \ldots, m_i. \qquad (2.2)$$

We develop a filtering method to cluster functional data that follow model (2.1). We first approximate the trajectories η_i from its noisy observations $\{Y_{ij} : j = 1, \ldots, m_i\}$ with a set of basis functions such as B-spline basis functions or estimated FPCs through the method of the principal components analysis through conditional expectation (PACE)

proposed by Yao et al. (2005). Let $\hat{\boldsymbol{\beta}}_i$ denote the estimated basis expansion coefficients from the B-spline representation. Suppose that the underlying true trajectory $\eta_i(x)$ can be approximated by these basis functions accurately, and that the corresponding coefficient vectors $\boldsymbol{\beta}_i$'s can be effectively leveraged to cluster these curves. For densely observed functional data, if m_i is sufficiently large for $i = 1, \ldots, n$, $\hat{\boldsymbol{\beta}}_i$ has appealing theoretical properties in terms of curve recovery. This explains why employing $\boldsymbol{\beta}_i$ as a proxy for η_i in clustering is reasonable. The FPC scores play a similar role as $\hat{\boldsymbol{\beta}}_i$'s. Similar ideas of using these coefficients for clustering functional data have been explored in Abraham et al. (2003), Jacques and Preda (2014), Wang et al. (2016) and references therein.

2.1 Curve recovery

In this subsection we review two methods of recovering curves: B-spline basis functions for densely observed functional data, and PACE for sparsely observed functional data.

2.1.1 Pre-determined basis expansion for densely observed functional data

For densely observed functional data, we utilize B-spline basis functions to recover the trajectory of $\eta_i(x)$ from the noisy observations Y_{ij}'s. This method has been widely adopted in FDA; refer to Ramsay and Silverman (2005) for a more detailed introduction. Denote by $\{\tau_J\}_{J=1}^{N_s}$ a sequence of equally-spaced points on $[0,\ 1]$, i.e., $\tau_J = J/(N_s + 1)$, $J \in \{1, \ldots, N_s\}$, $0 < \tau_1 < \cdots < \tau_{N_s} < 1$. They divide the interval $[0, 1]$ into $(N_s + 1)$ non-overlapping subintervals $\mathcal{I}_0 = (0, \tau_1]$, $\mathcal{I}_J = (\tau_J, \tau_{J+1}], J = 1, \cdots, N_s - 1$, $\mathcal{I}_{N_s} = (\tau_{N_s}, 1]$, and are often referred to as interior knots. For any positive integer p, let $\tau_{1-p} = \cdots = \tau_0 = 0$ and $1 = \tau_{N_s+1} = \cdots = \tau_{N_s+p}$ be auxiliary knots. Let $\mathcal{H}^{(p-2)} = \mathcal{H}^{(p-2)}[0, 1]$ be the polynomial spline space of order p, which consists of all

$(p-2)$ times continuously differentiable functions on $[0,1]$ that are polynomials of degree at most $(p-1)$ on each subinterval $\mathcal{I}_J$, $J \in \{0,\ldots,N_s\}$. Following De Boor (1978), we denote by $\{B_{J,p}(x), 1 \le J \le N_s+p\}$ the pth order B-spline basis functions of $\mathcal{H}^{(p-2)}$, and hence $\mathcal{H}^{(p-2)} = \left\{ \sum_{J=1}^{N_s+p} \beta_{J,p} B_{J,p}(x) \middle| \beta_{J,p} \in \mathbb{R}, x \in [0,1] \right\}$. Under the assumption that $\eta(x) \in \mathcal{H}^{(p-2)}$, the i-th unknown trajectory $\eta_i(x)$ is then estimated by:

$$\widehat{\eta}_i(\cdot) = \underset{f_i(\cdot)\in\mathcal{H}^{(p-2)}}{\operatorname{argmin}} \sum_{j=1}^{m_i} \{Y_{ij} - f_i(j/m_i)\}^2, i = 1,2,\cdots,n. \tag{2.3}$$

According to (2.3), one obtains

$$\widehat{\eta}_i(x) = \sum_{J=1}^{N_s+p} \widehat{\beta}_{i,J,p} B_{J,p}(x), \tag{2.4}$$

where $\widehat{\boldsymbol{\beta}}_i = \{\widehat{\beta}_{i,1,p}, \widehat{\beta}_{i,2,p}, \cdots, \widehat{\beta}_{i,N_s+p,p}\}^\top$ is defined as:

$$\widehat{\boldsymbol{\beta}}_i = \underset{\boldsymbol{\beta}_i\in\mathbb{R}^{N_s+p}}{\operatorname{argmin}} \sum_{j=1}^{m_i} \{Y_{ij} - \sum_{J=1}^{N_s+p} \beta_{i,J,p} B_{J,p}(j/m_i)\}. \tag{2.5}$$

Simple algebra yields that

$$\widehat{\boldsymbol{\beta}}_i = \left(\boldsymbol{X}_i^\top \boldsymbol{X}_i\right)^{-1} \boldsymbol{X}_i^\top \boldsymbol{Y}_i,$$

where $\boldsymbol{X}_i$ is the $m_i \times (N_s+p)$ design matrix,

$$\boldsymbol{X}_i = \begin{pmatrix} B_{1,p}(1/m_i) & \cdots & B_{N_s+p,p}(1/m_i) \\ \vdots & \vdots & \vdots \\ B_{1,p}(1) & \cdots & B_{N_s+p,p}(1) \end{pmatrix}$$

and $\boldsymbol{Y}_i = (Y_{i1}, Y_{i_2}, \cdots, Y_{im_i})^\top$. Ultimately, one obtains the coefficients matrix estimator

$\widehat{\boldsymbol{\Xi}}$ as follows:

$$\widehat{\boldsymbol{\Xi}} = \begin{pmatrix} \widehat{\boldsymbol{\beta}}_1^\top \\ \widehat{\boldsymbol{\beta}}_2^\top \\ \vdots \\ \widehat{\boldsymbol{\beta}}_n^\top \end{pmatrix} = \begin{pmatrix} \widehat{\beta}_{1,1,p}\widehat{\beta}_{1,2,p}\cdots\widehat{\beta}_{1,N_s+p,p} \\ \widehat{\beta}_{2,1,p}\widehat{\beta}_{2,2,p}\cdots\widehat{\beta}_{2,N_s+p,p} \\ \vdots \quad \vdots \quad \vdots \quad \vdots \\ \widehat{\beta}_{n,1,p}\widehat{\beta}_{n,2,p}\cdots\widehat{\beta}_{n,N_s+p,p} \end{pmatrix}.$$

2.1.2 Data-driven basis expansion for sparsely observed functional data

For sparsely observed functional data, one concern is that the presmoothing step is not reliable, as we cannot recover a curve accurately from only a few observations. Instead of using B-spline basis expansion coefficients, we use the FPC scores $\{\xi_{ik}\}$ in (2.1) for the subsequent analysis. In particular, we perform spectral clustering based on $\boldsymbol{\xi}_i = (\xi_{i1}, \cdots, \xi_{iK})^\top$, where K denotes the number of retained FPCs. To carry out FPC analysis from sparsely observed functional data, we adopt the PACE method proposed by Yao et al. (2005) to obtain the estimated FPC scores $\{\widehat{\xi}_{ik} : k = 1, \ldots, K, i = 1, \ldots, n\}$.

In PACE, local linear smoothers (Fan and Gijbels, 1996) are employed to obtain the estimated mean function $\widehat{\mu}(x)$ and covariance function $\widehat{G}(x, x')$. Then, according to

$$\int_0^1 \widehat{G}(x, x')\widehat{\phi}_k(x)dx = \widehat{\lambda}_k\widehat{\phi}_k(x'),$$

one obtains $\widehat{\phi}_k(x)$ and $\widehat{\lambda}_k$. Note that $E[\{Y_{ij} - \mu(X_{ij})\}\{Y_{ij'} - \mu(X_{ij'})\}] = G(X_{ij}, X_{ij'}) + \sigma_\epsilon^2\delta_{X_{ij},X_{ij'}}$, where $\delta_{X_{ij},X_{ij'}} = 1$ if $X_{ij} = X_{ij'}$ and 0 otherwise. One can estimate σ_ϵ^2 as $\widehat{\sigma}_\epsilon^2 = \int_0^1[\hat{V}(x) - \hat{G}(x, x)]dx$, where $\hat{V}(x)$ denotes the estimated variance function of $Y_i(x)$ by applying a local linear smoother to $\{(Y_{ij} - \hat{\mu}(X_{ij}))^2 : j = 1, \ldots, m_i, i = 1, \ldots, n\}$. Write $\widetilde{\boldsymbol{\eta}}_i = (\eta_i(X_{i1}), \cdots, \eta_i(X_{im_i}))^\top$, $\widetilde{\boldsymbol{Y}}_i = (Y_{i1}, \cdots, Y_{im_i})^\top$, $\boldsymbol{\mu}_i = (\mu(X_{i1}), \cdots, \mu(X_{im_i}))^\top$,

and $\boldsymbol{\phi}_{ik} = (\phi_k(X_{i1}), \cdots, \phi_k(X_{im_i}))^\top$. Then, under the normality assumption,

$$\widetilde{\xi}_{ik} = E[\xi_{ik}|\widetilde{\boldsymbol{Y}}_i] = \lambda_k \boldsymbol{\phi}_{ik}^\top \boldsymbol{\Sigma}_{Y_i}^{-1}(\widetilde{\boldsymbol{Y}}_i - \boldsymbol{\mu}_i),$$

provides a good approximation to the underlying true FPC scores, where $\boldsymbol{\Sigma}_{Y_i} = \text{cov}(\widetilde{\boldsymbol{Y}}_i, \widetilde{\boldsymbol{Y}}_i) = \text{cov}(\widetilde{\boldsymbol{\eta}}_i, \widetilde{\boldsymbol{\eta}}_i) + \sigma_\epsilon^2 \boldsymbol{I}_{m_i}$, i.e., the (p,q)th entry of $\boldsymbol{\Sigma}_{Y_i}$ is $G(X_{ip}, X_{iq}) + \sigma_\epsilon^2 \delta_{pq}$. By replacing these unknown quantities with their estimates, we obtain the estimated FPC scores

$$\widehat{\xi}_{ik} = \widehat{E}[\xi_{ik}|\widetilde{\boldsymbol{Y}}_i] = \widehat{\lambda}_k \widehat{\boldsymbol{\phi}}_{ik}^\top \widehat{\boldsymbol{\Sigma}}_{Y_i}^{-1}(\widetilde{\boldsymbol{Y}}_i - \widehat{\boldsymbol{\mu}}_i). \tag{2.6}$$

2.2 Functional Data Spectral Clustering

Recall that the basis coefficients $\{\widehat{\boldsymbol{\beta}}_i\}$ are derived from recovering trajectories from the noisy observations as in (2.4) under the dense condition, and the FPC scores $\{\widehat{\boldsymbol{\xi}}_i\}$ are derived from PACE as in (2.6) under the sparse condition. The use of $\widehat{\boldsymbol{\beta}}_i$ (or $\widehat{\boldsymbol{\xi}}_i$) enables the extension of multivariate clustering techniques to cluster functional data; see Wang et al. (2016) and Jacques and Preda (2014) for a more detailed discussion. While k-means is a classic approach for multivariate clustering (Abraham et al., 2003; Garcia-Escudero and Gordaliza, 2005), its reliance on convex Voronoi regions limits its effectiveness for convex datasets, rendering it unsuitable for non-convex data. In contrast, spectral clustering represents data as interconnected entities, employing edge weights based on proximity or similarity, making it more adaptable to diverse data distributions. As illustrated in Figure 1, spectral clustering demonstrates superior performance compared to traditional k-means, particularly for non-convex datasets. Given that the distribution of $\widehat{\boldsymbol{\beta}}_i$ (or $\widehat{\boldsymbol{\xi}}_i$) is typically unknown, we adopt spectral clustering in the second stage to cluster $\widehat{\boldsymbol{\beta}}_i$ (or $\widehat{\boldsymbol{\xi}}_i$) and map these clusters back to the original dataset. These constitute the two-stage

functional data spectral clustering (TSFDSC) method. To the best of our knowledge, few methods can simultaneously handle both sparse and dense functional data while accommodating distribution-free, non-convex coefficients that encapsulate clustering structures. TSFDSC addresses these challenges, offering an effective methodology. For brevity, we only demonstrate the second stage for densely observed functional data, i.e. using $\widehat{\boldsymbol{\beta}}_i$; for sparsely observed functional data, we only need to replace $\widehat{\boldsymbol{\beta}}_i$ by $\widehat{\boldsymbol{\xi}}_i$.

In spectral clustering, the weights ω_{ij} between $\boldsymbol{\beta}_i$ and $\boldsymbol{\beta}_j$ are first computed to form the weighted adjacency matrix $\boldsymbol{W} = (\omega_{ij})_{n\times n}$. The degree matrix $\boldsymbol{D}$ is then constructed as a diagonal matrix with elements $d_i \triangleq \sum_{j=1}^{n} \omega_{ij}$. Determining ω_{ij} poses a practical challenge, which we address by constructing a similarity matrix $\boldsymbol{S}$. Specifically, we adopt a fully connected graph where $\omega_{ij} > 0$ for all i, j and utilize the radial basis function (RBF) kernel to define the edge weights. Consequently, the similarity matrix $\boldsymbol{S} = (s_{ij})_{n\times n}$ is equivalent to $\boldsymbol{W}$, with $\omega_{ij} = s_{ij} = \exp\{-\|\boldsymbol{\beta}_i - \boldsymbol{\beta}_j\|^2/2\iota^2\}$, where ι controls the neighborhood width. The degree matrix $\boldsymbol{D}$ is subsequently obtained.

Remark 2.1. *Measuring similarity or distance between sample points in sparse functional data presents a significant challenge. However, TSFDSC offers an effective solution to this issue. The approach proceeds in two key stages. First, smoothing techniques are applied to the sparse observation points, transforming them into curves. This preprocessing step facilitates more accurate similarity measurement during the construction of the graph Laplacian. Second, the graph Laplacian is constructed using a smooth kernel function (Gaussian kernel), that operates on basis coefficients rather than directly on the sparse data points. This modification ensures a more robust similarity assessment, even in sparse data settings. The use of the Gaussian kernel further aids in mitigating the sparsity problem by adjusting distances to preserve connectivity and symmetry within the graph structure, thereby enhancing the overall effectiveness of the algorithm.*

The unnormalized graph Laplacian is defined as $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W}$, where $\boldsymbol{D}$ is the degree matrix, and $\boldsymbol{W}$ is the weighted adjacency matrix. The normalized graph Laplacian is then given by $\boldsymbol{L}_{\text{norm}} = \boldsymbol{D}^{-1/2}\boldsymbol{L}\boldsymbol{D}^{-1/2}$ (Grone et al., 1990; Chen et al., 2024). The first Q eigenvectors of $\boldsymbol{L}_{\text{norm}}$, denoted as $\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_Q$, are computed and normalized

to form the matrix $\boldsymbol{V} \in \mathbb{R}^{n\times Q}$, where each column represents a normalized eigenvector. Let $\boldsymbol{r}_i \in \mathbb{R}^Q$ denote the i-th row of $\boldsymbol{V}, i = 1, \cdots, n$. The rows $\{\boldsymbol{r}_i\}_{i=1,\cdots,n}$ are then partitioned into Q optimal clusters $C_1^\star, \cdots, C_Q^\star$ using k-means clustering. By rearranging the subscript, the corresponding optimal clusters of the functional coefficients $\boldsymbol{\beta}_i$ are denoted as $SC_1^\star, \cdots, SC_Q^\star$, where $SC_q^\star = \{\boldsymbol{\beta}_i | \boldsymbol{r}_i \in C_q^\star\}$, meaning that if $\{\boldsymbol{r}_i\} \in C_q^\star$, then $\boldsymbol{\beta}_i \in SC_q^\star$. Consequently, the optimal clusters of the underlying trajectories are represented as $DSC_1^\star, \cdots, DSC_Q^\star$ where $DSC_q^\star = \{\eta_i | \boldsymbol{\beta}_i \in SC_q^\star\}$. A detailed outline of the TSFDSC algorithm is provided in Algorithm 1 in which $\widehat{DSC}_q^\star$, $\widehat{SC}_q^\star$ and $\widehat{C}_q^\star$ are optimal clusters of any $\widehat{DSC}_q$, $\widehat{SC}_q$, $\widehat{C}_q$, calculated based on $\{Y_{ij}, j = 1, \ldots, m_i, i = 1, \ldots, n\}$.

Algorithm 1 Two Stage Functional Data Spectral Clustering (TSFDSC)

Input: observations $Y_{ij}, i = 1, 2, \cdots, n, j = 1, 2, \cdots, m_i$, number of clusters to construct.
if Y_{ij} are dense
  Input: the number of knots N_s and order p for B-spline,
  use B-spline to recover $\{\eta_i(t)\}_{i=1}^n$ with order p and knots number N_s
  obtain basis coefficients $\{\widehat{\boldsymbol{\beta}}_i\}_{i=1}^n$
elif Y_{ij} are sparse
  Input: the number of retained FPCs K
  obtain FPC scores $\{\widehat{\boldsymbol{\xi}}_i\}_{i=1}^n$ through PACE
use fully connected graph to construct the similarity matrix $\widehat{\boldsymbol{S}}$
according to the similarity matrix $\widehat{\boldsymbol{S}}$, construct the weighted adjacency matrix $\widehat{\boldsymbol{W}}$ and degree matrix $\widehat{\boldsymbol{D}}$
compute Laplacian matrix $\widehat{\boldsymbol{L}} = \widehat{\boldsymbol{D}} - \widehat{\boldsymbol{W}}$
construct the normalized Laplacian matrix $\widehat{\boldsymbol{L}}_{\text{norm}} = \widehat{\boldsymbol{D}}^{-1/2}\widehat{\boldsymbol{L}}\widehat{\boldsymbol{D}}^{-1/2}$
compute the first Q eigenvectors $\widehat{\boldsymbol{v}}_1, \widehat{\boldsymbol{v}}_2, \cdots, \widehat{\boldsymbol{v}}_Q$ of $\widehat{\boldsymbol{L}}_{\text{norm}}$
form matrix $\widehat{\boldsymbol{V}} \in \mathbb{R}^{n\times Q}$ and compute its normalized eigenvectors $\widehat{\boldsymbol{v}}_1, \widehat{\boldsymbol{v}}_2, \cdots, \widehat{\boldsymbol{v}}_Q$
denote the rows of $\widehat{\boldsymbol{V}}$ as clusters $\widehat{\boldsymbol{r}}_i, i = 1, 2, \cdots, n$
apply k-means to each cluster $\widehat{\boldsymbol{r}}_i$, obtain the clustering results $\widehat{C}_1^\star, \widehat{C}_2^\star, \cdots, \widehat{C}_Q^\star$
obtain $\widehat{SC}_1^\star, \widehat{SC}_2^\star, \cdots, \widehat{SC}_Q^\star$ and corrospondingly $\widehat{DSC}_1^\star, \widehat{DSC}_2^\star \cdots, \widehat{DSC}_Q^\star$
Output: clusters $\widehat{DSC}_1^\star, \widehat{DSC}_2^\star, \cdots, \widehat{DSC}_Q^\star$

2.3 Theoretical Results

In this section we investigate the large sample properties of the proposed procedure for densely observed functional data.

The Laplacian matrix is essential for partitioning a graph in spectral clustering, akin to clustering in graph theory. The objective is to partition the graph such that the sum of edge weights within subgraphs is maximized while minimizing those between subgraphs. For two disjoint clusters SC_q and $SC_{q'}$, we define the cut as $\text{cut}(SC_q, SC_{q'}) = \sum_{\boldsymbol{\beta}_i \in SC_q, \boldsymbol{\beta}_j \in SC_{q'}} \omega_{ij}$. Extending this to multiple clusters, the total cut is expressed as $\text{cut}(SC_1, SC_2, \cdots, SC_Q) = \sum_{q=1}^{Q} \text{cut}(SC_q, \overline{SC_q})$, where $\overline{SC_q}$ denotes the complement of cluster q (chapter 8 of Aggarwal and Reddy (2018)). The standard approach to graph partitioning involves solving the minimum cut problem: $\min \text{cut}(SC_1, SC_2, \cdots, SC_Q)$. Spectral clustering, however, also ensures that each subgraph contains a sufficiently large number of vertices. To improve partitioning, more refined methods impose additional constraints. For instance, RatioCut seeks to maximize the number of points per subgraph by minimizing the loss function $\sum_{q=1}^{Q} \text{cut}(SC_q, \overline{SC_q})/|SC_q|$. Similarly, NCut (Normalized Cut) aims to maximize the sum of vertex degrees within each subgraph, using the objective $\mathcal{N}_{\text{cut}}(SC_1, SC_2, \cdots, SC_Q) = \sum_{q=1}^{Q} \text{cut}(SC_q, \overline{SC_q})/\text{vol}(SC_q)$, where $\text{vol}(SC_q) = \sum_{\boldsymbol{\beta}_i \in SC_q} d_i$. Suppose there exists an optimal partition $\{SC_q^\star, q = 1, 2, \cdots, Q\}$ determined by $\{\boldsymbol{\beta}_i, i = 1, 2, \cdots, n\}$ that minimises $\mathcal{N}_{\text{cut}}(SC_1, SC_2, \cdots, SC_Q)$. The corresponding clusters are denoted as $\{C_q^\star, q = 1, 2, \cdots, Q\}$. Despite the shared goals, both RatioCut and NCut are NP-hard problems, requiring approximation methods for effective solutions.

On the one hand, for the 2-clustering problem ($Q = 2$), the objective function can be transformed from a constrained optimization problem to an unconstrained one using relaxation techniques and the Lagrange multiplier method. It is found that the minimum value of this objective function relates to the eigenvalues of the Laplacian matrix. Specifically, the smallest eigenvalue of the Laplacian matrix is always zero. As shown by Von Luxburg (2007), the second smallest eigenvalue, known as the Fiedler value, corre-

sponds to the optimal value of the objective function, and the corresponding eigenvector (Fiedler vector) provides the optimal solution for partitioning. On the other hand, for Q-clustering ($Q > 2$), approximate solutions are employed. In this case, the first Q eigenvalues of the Laplacian matrix are considered, and the corresponding eigenvectors are assembled into a matrix. This matrix is then clustered using the k-means algorithm, with the clustering results mapped back to the original data. The eigenvalues of the Laplacian matrix represent the cost of cutting the graph, while the associated eigenvectors guide the partitioning process, ensuring an efficient and effective clustering.

Next, we develop the asymptotic properties of TSFDSC under the following technical assumptions. For a non-negative integer p and a real number $\gamma \in (0, 1]$, define the space of γ-Hölder continuous functions as

$$\boldsymbol{\mathcal{H}}^{(p,\gamma)}[0,1] = \left\{ \kappa : [0,1] \to \mathbb{R} \middle| \, \|\kappa\|_{p,\gamma} = \sup_{x,y\in[0,1],x\neq y} \left| \frac{\kappa^{(p)}(x) - \kappa^{(p)}(y)}{|x-y|^{\gamma}} \right| < +\infty \right\}.$$

Assumption 2.1. There exist an integer $p > 0$ and a constant $\gamma \in (0, 1]$, such that $\eta_i(x) \in \boldsymbol{\mathcal{H}}^{(p,\gamma)}[0, 1]$, $i = 1, 2, \cdots, n$.

Assumption 2.2. In model (2.2), ϵ_{ij}'s are i.i.d copies of ϵ. Moreover, for any $r \geq 3$, $\mathrm{E}\,|\epsilon|^r < \infty$.

Assumption 2.3. The clusters $\{SC_q^\star, q = 1, 2, \cdots, Q\}$ are unique. Similarly, the k-means clusters derived from $\boldsymbol{V}$, denoted as $\{C_q^\star, q = 1, 2, \cdots, Q\}$ are also unique. $\{\widehat{SC}_q^\star, q = 1, 2, \cdots, Q\}$ and $\{\widehat{C}_q^\star, q = 1, 2, \cdots, Q\}$ are also uniquely determined.

Assumption 2.4. The similarity function is bounded away from zero by a positive constant, i.e., there exists a constant $0 < c < 1$, such that $\omega_{ij} = \exp\{-\|\boldsymbol{\beta}_i - \boldsymbol{\beta}_j\|^2/2\iota^2\} > c$ for all i, j.

Assumption 2.5. Let $m = \min\{m_1, m_2, \cdots, m_n\}$, as $n, m \to +\infty$, $N_s^3 n^2 m^{-1} \log m = o(1)$.

Assumption 2.1 controls the smoothness of the trajectory function by imposing a Hölder condition on its derivative. This assumption is often adopted when using B-spline basis functions to approximate an unknown function. The moment condition for ϵ_{ij} in Assumption 2.2 implies that $\{m_i^{-1} B_{J,p}(j/m_i)\,\epsilon_{ij} : j = 1, \ldots, m_i\}$ satisfies Cramer's conditions. Hence we can apply Bernstein's inequality to the error term to achieve the convergency of the estimated coefficients (details can be found in the supplementary file). Assumption 2.3 ensures a unique partitioning of the coefficient vector into clusters SC_q or $SC_{q'}$, $q \neq q'$ that minimizes the objective function. This relates to sparse clustering, which seeks groups of variables with similar effects on the response variable. Assumption 2.4 concerns properties of the similarity function in spectral clustering. Von Luxburg et al. (2008) pointed out that the similarity function should be symmetric, continuous, and bounded away from zero by a positive constant to form an undirected graph, ensuring small data changes do not significantly affect results. Such properties are desirable to ensure robustness of spectral clustering. Since we use the Gaussian kernel to define the similarity function, only Assumption 2.4 is required to achieve these properties. Assumption 2.5 describes the sampling frequency of noisy observations. The following theorem establishes the convergence rate of the estimated spline coefficients.

Theorem 2.1. Under Assumptions 2.1 - 2.2, as $m \to \infty$, one has

$$\max_{1 \le i \le n} \|\widehat{\boldsymbol{\beta}}_i - \boldsymbol{\beta}_i\|^2 = O_p(N_s^3 m^{-1} \log m).$$

Recall that $\mathcal{N}_{\text{cut}}(SC_1, SC_2, \cdots, SC_Q) = \sum_{q=1}^{Q} \text{cut}(SC_q, \overline{SC_q}) / \text{vol}(SC_q)$ defines the loss function of clustering. We further show that the loss function of our proposed clus-

tering algorithm converges to that of the optimal clustering based on the $\boldsymbol{\beta}_i$'s.

Theorem 2.2. Under Assumptions 2.1-2.5, one obtains that

$$\lim_{n\to\infty}\left|\mathcal{N}_{\text{cut}}(\widehat{SC}_1^\star,\widehat{SC}_2^\star,\cdots,\widehat{SC}_Q^\star)-\mathcal{N}_{\text{cut}}(SC_1^\star,SC_2^\star,\cdots,SC_Q^\star)\right|=0, a.s.$$

where the set $\{SC_i^\star\}_{q=1}^Q$ and $\{\widehat{SC}_i^\star\}_{q=1}^Q$ is the minimizer of $\mathcal{N}_{\text{cut}}(SC_1,SC_2,\cdots,SC_Q)$ and $\mathcal{N}_{\text{cut}}(\widehat{SC}_1,\widehat{SC}_2,\cdots,\widehat{SC}_Q)$, respectively.

Remark 2.2. To establish theoretical properties of the estimator under the sparse setting, we first need to find the convergence rate of the estimated FPC scores, as in Theorem 2.1 for dense functional data. Yao et al. (2005) established the convergence rate of the estimated FPCs in their Theorem 2. But they only showed that $\hat{\xi}_{ik}$'s converge to $\tilde{\xi}_{ik}$ in probability in Theorem 3. There are two major limitations with this result. First, they did not establish the consistency of the estimated FPC scores as an estimate of ξ_{ik}. Second, the convergence rate is unknown. To our best knowledge, the convergence rate of the estimated FPC scores through the PACE method (Yao et al., 2005) has not been established in the literature. Most literature has been focused on the theoretical properties of the estimated mean function, covariance function and eigenfunctions for sparse functional data. For example, Zhang and Wang (2016) developed the L^2 convergence rate, uniform convergence rate and the asymptotic normality of the estimated mean and covariance functions under various sampling frequencies, while Li and Hsing (2010) established the uniform convergence rate for the estimated mean function, covariance function, eigenfunctions, and eigenvalues. Both limitations hinder the theoretical development of clustering based on $\hat{\xi}_{ik}$'s under the sparse setting. If we follow the strategy in Zhou et al.

(2023) to define the estimated FPC score as

$$\hat{\xi}_{ik} = \int_0^1 \{\eta_i(t) - \hat{\mu}(t)\}\hat{\phi}_k(t)dt \approx \frac{1}{m_i}\sum_{j=1}^{m_i}\{Y_{ij} - \hat{\mu}(X_{ij})\}\hat{\phi}_k(X_{ij}),$$

dense observations are needed to ensure that the Riemann sum can provide a good approximation to the integral.

Another critical issue in the theoretical development is that $\hat{\xi}_{ik}$'s are not independent between these subjects, because we employ observations from all subjects to estimate the FPCs. A possible solution to this dependency issue is to use the sample split method as in Zhou et al. (2023). However, this method entails dense functional data as well.

2.4 Implementation

In this subsection, we propose two methods to select the number of knots N_s. First, based on Wang et al. (2020) and Ma (2014), we suggest using the empirical formula $N_s = c\{n^{1/(2p)}\}^{\rho}\{\text{loglog}(n)\}^{\rho-1}$, with $c = 14$ and $\rho = 3/8$ as recommended. Second, we consider a BIC-based selection strategy, where N_s is chosen from $\left[\lfloor 3n^{1/(2p+1)}\rfloor, \lfloor 10n^{1/(2p+1)}\rfloor\right]$, with $\lfloor\cdot\rfloor$ denoting the floor function. To determine the optimal parameter ι in the RBF kernel $\widehat{\omega}_{ij} = \exp\{-\|\widehat{\boldsymbol{\beta}}_i - \widehat{\boldsymbol{\beta}}_j\|^2/2\iota^2\}$, we adopt a heuristic approach. In particular, for each candidate ι, spectral clustering is implemented, and finally the ι with the smallest sum of distances within the clusters is selected. This procedure can be implemented through the specc function in the R package kernlab.

3. Simulation Studies

3.1 Data generating process

To evaluate the performance of TSFDSC, we conduct simulations in this section and compare it with the FD k-means clustering, which uses k-means to cluster the coefficients directly. In each simulation replicate, the repeated measures for each subject are generated as follows:

$$Y_{ij} = \eta(t_{ij}) + \epsilon_{ij} = \mu(t_{ij}) + \sum_{k=1}^{K} \xi_{ik}\phi_k(t_{ij}) + \epsilon_{ij}, \quad i = 1, \cdots, n, j = 1, \cdots, m_i,$$

in which the noise ϵ_{ij} is i.i.d. from $N(0, \sigma^2)$, the FPC scores $\xi_{ik} \sim N(0, \zeta_k)$, and $\phi_k(t)$ denotes the corresponding FPC. We consider the following five settings for the observation times.

- (1) **Dense settings**

  In **setting 1.a**, all curves have $m_i = m \in \{21, 51, 101\}$ observation points and $t_{ij} = j/m$ for $j = 1, \ldots, m_i$.

  In **setting 1.b**, the number of observations made on each curve, m_i, is randomly selected from $\{21, \cdots, 101\}$ and $t_{ij} = j/m_i$ for $j = 1, \ldots, m_i$.

- (2) **Regular sparse settings**

  In **setting 2.a**, all curves have $m_i = m \in \{6, 11\}$ observation points and $t_{ij} = j/m$ for $j = 1, \ldots, m_i$.

  In **setting 2.b**, the number of observations made on each curve, m_i, is randomly selected from $\{4, \cdots, 11\}$ or $\{4, \cdots, 21\}$ and $t_{ij} = j/m_i$ for $j = 1, \ldots, m_i$. The latter case where the number of observations on each curve follows a uniform distribution

on $\{4, \cdots, 21\}$ can be treated as a mixed type where some trajectories have relatively dense observations while others have sparse observations.

- (3) **Irregular sparse setting**

  In **setting 3**, following Yao et al. (2005), we first define an equally spaced grid $\{\tau_0, \cdots, \tau_{20}\}$ on $[0, 1]$ with $\tau_0 = 0, \tau_{20} = 1$. Then for the ith trajectory, let $s_i = \tau_i + e_i$, where e_i are i.i.d. with $N(0, 0.1)$, $s_i = 0$ if $s_i < 0$ and $s_i = 1$ if $s_i > 1$. The number of observations, m_i, is randomly selected from $\{1, 2, 3, 4\}$. To avoid repeated selection of 0 or 1, t_{ij}'s are randomly chosen from the set of s_i's with 0 and 1 excluded.

Regarding the mean function, the eigenvalues and the eigenfunction of the covariance function of η, we consider the following designs. We choose $K = 2$ as recommended in Wang et al. (2022) and set eigenfunctions and eigenvalues as

$$\phi_1^{(1)}(x) = \sqrt{2}\sin(0.5\pi x), \phi_2^{(1)}(x) = \sqrt{2}\cos(0.5\pi x), \text{ and } \sqrt{\zeta^{(1)}} = (0.4, 0.35)^\top,$$
$$\phi_1^{(2)}(x) = \sqrt{2}\sin(\pi x), \phi_2^{(2)}(x) = \sqrt{2}\cos(\pi x), \text{ and } \sqrt{\zeta^{(2)}} = (0.3, 0.25)^\top,$$
$$\phi_1^{(3)}(x) = \sqrt{2}\sin(2\pi x), \phi_2^{(3)}(x) = \sqrt{2}\cos(2\pi x), \text{ and } \sqrt{\zeta^{(3)}} = (0.2, 0.15)^\top.$$

The mean functions are specified as $\mu_1(x) = 4(x - 0.5)^2 + 2$, $\mu_2(x) = -2(x - 0.5)^2 + x$ and $\mu_3(x) = 2.5e^{-25(x-0.25)^2} + 2e^{-50(x-0.75)^2}$. Plots of these mean functions are presented in Figure S.1 of the supplementary file, where $\mu_1(x)$ exhibits a symmetric U-shape, $\mu_2(x)$ displays an asymmetric inverted U-shape, and $\mu_3(x)$ follows a zigzag pattern, capturing various trends. Moreover, we set $\sigma \in \{0.01, 0.3\}$ and sample size $n \in \{50, 100, 200\}$. We consider six scenarios in total in terms of the specification of the mean function and the covariance operator, and each scenario consists of three clusters. The details of these six scenarios are summarized in Table 1.

For dense settings, trajectories are recovered using cubic splines with $p = 4$, and the number of knots is determined via the empirical formula method and BIC, as described in Section 2.4. To assess sensitivity, we employ both adaptive selection of ι using the heuristic approach outlined in Section 2.4, and fixed values $\iota = 0.1, 0.5, 0.9$, chosen based on prior experience. For regular sparse settings, since the observation points we set is at least 4, we also use B-spline coefficients to perform clustering.

3.2 Normalized Mutual Information

Traditional external measures in cluster analysis include Accuracy (Ac), defined as the maximal possible ratio of correctly classified objects to the total number of objects to be clustered, or, for simplicity, the total correct outcomes among the total outcomes made. Denote by $\{x_i\}_{i=1}^n$ the sample, $\mathcal{T}$ the true class labels, l_t the number of true classes, $\mathcal{T}_i$ the true class i, $\mathcal{T}(x_i)$ the true label of x_i. Let $\mathcal{C}$ denote the given clustering result, l_c the number of clusters in $\mathcal{C}$, and $\mathcal{C}(x_i)$ the result label of x_i. Then Ac is calculated as $\text{Ac} = n^{-1}\sum_{i=1}^n \delta_{\mathcal{T}(x_i),map(\mathcal{C}(x_i))}$, where $map(\cdot)$ denotes the optimal reassignment (one to one) of result labels, $\delta_{\mathcal{T}(x_i),map(\mathcal{C}(x_i))} = 1$ if $\mathcal{T}(x_i) = map(\mathcal{C}(x_i))$ and 0 otherwise.

However, Ac fails to account for the number of clusters, as it does not penalize solutions with an excessive number of clusters. Meanwhile, Ac is only valid when $l_t = l_c$. When $l_t \neq l_c$ (over-clustering/under-clustering), Ac would be severely distorted. In contrast, Normalized Mutual Information (NMI) (Kachouie and Shutaywi, 2020) addresses this issue by considering both the clustering structure and its alignment with true labels.

Mutual Information (MI) is defined by $\text{MI} = H(\mathcal{T}) - H(\mathcal{T}|\mathcal{C})$, where

$$\begin{aligned} H(\mathcal{T}) &= -\sum_{i=1}^{l_t} P(\mathcal{T}_i)\log\left(P(\mathcal{T}_i)\right), P(\mathcal{T}_i) = \frac{|\mathcal{T}_i|}{|\mathcal{T}|}, \\ H(\mathcal{T}|\mathcal{C}) &= -\sum_{j=1}^{l_c} P(\mathcal{C}_j)\sum_{i=1}^{l_t} P(\mathcal{T}_i|\mathcal{C}_j)\log\left(P(\mathcal{T}_i|\mathcal{C}_j)\right), P(\mathcal{T}_i|\mathcal{C}_j) = \frac{|\mathcal{T}_i \cap \mathcal{C}_j|}{|\mathcal{C}_j|}. \end{aligned}$$

NMI is then given by NMI $= [\max\{H(\mathcal{T}), H(\mathcal{C})\}]^{-1}$ MI, where $H(\mathcal{C})$ is defined analogously to $H(\mathcal{T})$.

NMI accounts for the number of clusters by incorporating entropy in the denominator, which increases as the number of clusters grows. This causes the NMI to drop when the clustering outcome loses meaningfulness, thus effectively penalizing excessive partitioning. In contrast, when $l_c > l_t$, Ac might be high even when the clustering performance is poor. NMI, however, reflects the true quality by returning a low value, offering a more reliable evaluation. For example, consider $\mathcal{T}_1 = \{x_i\}_{i=1}^{50}$, $\mathcal{T}_2 = \{x_i\}_{i=51}^{100}$, $\mathcal{C}_1 = \{x_i\}_{i=31}^{90}$, $\mathcal{C}_2 = \{x_i\}_{i=1}^{30}$, $\mathcal{C}_3 = \{x_i\}_{i=91}^{100}$. In this case, NMI $= 0.35$ while Ac $= 0.7$. While for $\mathcal{C}_1 = \{x_i\}_{i=1}^{30} \cup \{x_i\}_{i=91}^{100}$, $\mathcal{C}_2 = \{x_i\}_{i=31}^{90}$, NMI $= 0.12$ while Ac $= 0.7$. Although Ac appears high, the clustering result is unreliable, a fact correctly captured by the low NMI value. While Ac may be large, NMI offers a more refined measure by capturing the shared information between clusters and true labels. Additionally, NMI is highly sensitive when the clustering accuracy exceeds 80%, with small changes in accuracy leading to significant variations in the NMI score (Kachouie and Shutaywi, 2020).

3.3 Simulation Results

The performance of the proposed TSFDSC and FD k-means is evaluated by clustering trajectories into three categories based on B-spline smoothing coefficients or FPC scores over 100 replications. The distanced-based functional spectral clustering approach (FSC-S) of Al Alawi (2021) and kCFC of Chiou and Li (2007) are also considered in **setting 1.a**. Figure S.2 in the supplementary file displays the recovered trajectories from the B-spline smoothing for Scenarios 1, 3, and 5 from a single replication. In Scenario 1, although the mean curves of these three clusters are identical, the eigenfunctions and eigenvalues of their covariance operators are different, thus showing different variation patterns. In

contrast, for Scenarios 3 and 5, where the mean curves differ, the smoothed trajectories exhibit distinct patterns, clearly separating the clusters.

To save space, Table 2 shows some results regarding the performance of TSFDSC with adaptive ι, FD k-means, FSC-S(D_0) (distance of original curves), FSC-S(D_1) (distance of first-order derivatives of curves), FSC-S(D_2) (distance of second-order derivatives of curves) and kCFC; full results can be found in Table S.1 of the supplementary file. Table 3 shows the performance of our methods and FD k-means clustering under settings 1.a and 2.a, and Table 4 shows their performance under settings 1.b and 2.b. The results of BIC-based selected N_s can be found in Tables S.3 - S.4 of the supplementary file. Moreover, Tables 5 and 6 show the performance when FPC scores are used in TSFDSC and FD k-means under regular and irregular sparse settings.

Tables 2 and S.1 indicate that the performance of FSC-S and kCFC is generally inferior to that of FD k-means and TSFDSC, especially in Scenarios 3 and 4. Moreover, FSC-S proposed in Al Alawi (2021) is not applicable to irregular or sparse functional data. Therefore, we consider them only in **setting 1.a**. For kCFC, some clusters may contain only one curve during the initial process, making it infeasible to perform FPCA, and thus we exclude it in Scenarios 1 and 2.

Table 3 summarizes results regarding the clustering performance of TSFDSC using both adaptive and empirical selection of ι, and the full results can be found in Table S.2 of the supplementary file. TSFDSC with empirical ι achieves the best performance in most cases, while the adaptive selection of ι ranks second, with 25 out of 90. However, the performance of TSFDSC with different ι's varies greatly and the empirical selection of ι requires multiple rounds of experimentation, leading to increased computational and experimental costs. In terms of the negligible difference observed between the best empirical and adaptive selections, the adaptive selection of ι is recommended for practical

use. Accordingly, only the results from this adaptive selection are presented in Table 4 and Tables S.3 - S.4 of the supplementary file for the TSFDSC method.

We examine the performance of our method based on different N_s, the number of B-spline basis functions used in recovering trajectories. Tables 3 and 4 present the results for the empirical formula-based N_s. For the setting of a fixed number of observations per curve, Table 3 shows that these two methods have relatively small NMI in Scenarios 1 and 2, where three clusters share the same mean function but differ in the covariance function. Combining this with the performance under Scenarios 1 and 2 in other tables, we hold the view that for filtering methods, the mean function is the main factor that affects the clustering results. Therefore, for Scenarios 1 and 2 where different clusters share the same mean curves but have different variations, TSFDSC and FD k-means do not work well. In contrast, for Scenarios 3 - 6, different clusters have different mean functions. As expected, both methods perform reasonably well in these scenarios. However, TSFDSC outperforms FD k-means in Scenarios 3 and 4, where the mean curves have similar shapes but differ by small numerical offsets (constant differences). In Scenarios 5 and 6 where different clusters are characterized by different mean functions in both shape and magnitude, TSFDSC slightly outperforms FD k-means, although their performance becomes comparable as both m_i and n increase. As the noise level increases, both methods exhibit a reduced NMI, especially in Scenario 4, where small n and m_i lead to decreased cluster separability, which is expected given that a higher noise level typically undermines clustering performance.

We also consider the performance of these two methods based on the BIC-based selected N_s, where the results are summarized in Tables S.3 - S.4 of the supplementary file. Compared with those presented in Tables 3 and 4 with empirical formula-based N_s, we find that the BIC-based selected N_s does not significantly enhance clustering performance, with the exception of FD k-means in dense settings. Moreover, calculating

BIC for each candidate N_s is computationally more expensive than the empirical formula-based method, especially for large sample sizes. In Scenarios 5 and 6, both TSFDSC and FD k-means achieve comparable performance, with NMI values exceeding 0.8, indicating over 90% correct clustering (Kachouie and Shutaywi, 2020). The NMI difference between the two methods is typically less than 0.05, occasionally reaching 0.1 in extreme cases, meaning that their performances are virtually indistinguishable. Notably, TSFDSC shows minimal variation (less than 0.03) in NMI and Ac between the BIC-based and empirical formula-based knot selection, highlighting the robustness of TSFDSC. From Tables 3 - 4 and Tables S.3 - S.4, we find that TSFDSC outperforms the traditional FD k-means method for dense functional data.

Table 5 summarizes the clustering performance of TSFDSC and FD k-means for regular sparse functional data. With the increase of n and m_i, the clustering results become more accurate and TSFDSC always outperforms FD k-means. However, for irregular sparse or extremely sparse (only one observation) functional data, implementing B-spline smoothing may not be practical, so we only show the performance of TSFDSC and FD k-means using FPC scores, as shown in Table 6. Since the NMI of scenarios 1-6 is relatively small, we also show their Ac in parentheses in Table 6. In general, TSFDSC performs relatively well for big noise levels, roughly guaranteeing an accuracy of 65%-75%. Compared with FD k-means, the two have their own advantages and disadvantages.

Overall speaking, our proposed performs reasonably well under various simulation designs.

4. Real Data Analysis

In the following subsections, TSFDSC is applied to the Berkeley Growth data and CD4 data.

4.1 Berkeley Growth data

The Berkeley Growth data was collected from a long-term study conducted by the California Institute of Child Welfare, focusing on child development. This dataset includes height values over time. It consists of height measurements at 31 time points for 93 children, aged 1 to 18, including 39 boys and 54 girls. Heights were measured quarterly from age 1 to 2, annually from age 2 to 8, and biannually from age 8 to 18. The dataset is available from the R package `fda` (Ramsay, 2024).

The adolescent growth spurt is a period of rapid increase in height and weight. According to Bogin (1999) and Kronenberg (2007), it generally begins around ages 9 - 10 for females and 11 - 12 for males. While males start later, their growth lasts longer (ages 10 - 15) and peaks around age 14. In contrast, females experience shorter spurts (ages 9 - 13) and peak around age 11. Hence females are usually taller than males between 10 and 13, but the opposite is true after 15, and males are generally taller than females in adulthood.

Given these growth patterns, traditional clustering methods may struggle to capture the complexities involved. Since the distribution of the potential trajectory coefficients (of recovering height curves) is complex and unknown, we first apply TSFDSC to cluster these height curves and evaluate its effectiveness in identifying gender differences, with gender serving as the ground truth label.

The height data are fitted using cubic splines (order $p = 4$), with the recovered height curves depicted in panel (a) of Figure 2. Table 7 summarizes the clustering performance of multiple approaches in the height growth data. The comparison includes FD k-means, our TSFDSC method under two representations, i.e., PACE FPC scores (TSFDSC(PACE)) and B-spline coefficients (TSFDSC(Bs)), as well as k-means based on conventional FPC scores (k-means(cFPC)) and spectral clustering based on cFPC (SC(cFPC)). Results

from relevant literature (Delaigle et al., 2019; Al Alawi, 2021; Chiou and Li, 2007) are also included. Notably, the TSFDSC(Bs) method shows favorable performance, achieving the best results among all eight methods compared.

As we have described before, the adolescent growth spurt causes the difference in height curves between males and females. The fundamental reason is the difference in growth rates in various aspects. Females' adolescent growth spurt comes earlier than males', but lasts shorter than males'. Meanwhile, the peak growth rate of females during the adolescent growth spurt is also lower than that of males (Largo et al., 1978; Bogin, 1999). Growth is defined as the difference in height between consecutive measurements, derived from the original height data. For the growth curves, quartic splines (order $p = 5$) are used, as illustrated in panel (b) of Figure 2. Floriello and Vitelli (2017) focused on growth curves and performed clustering, and employed a confusion matrix to display cluster results. As shown in Table 7, TSFDSC(Bs) also achieves the best performance in clustering growth curves.

Figure 3 shows that cluster 1 (male) displays a noticeable growth surge after age 13. In contrast, cluster 2 (female) experiences growth spurts around ages 4 and 10, reflecting the pattern that females typically enter their adolescent growth spurt earlier than males. Notably, while many females stop growing taller by ages 15 or 16, some males continue to experience growth spurts. These findings suggest that females are generally taller than males until around age 12, but by age 14, males typically surpass females in height, as shown by the clustering results. TSFDSC allows for an in-depth analysis of the spectral characteristics of these growth curves, revealing latent growth patterns and providing accurate classifications. This approach not only highlights individual growth differences but also offers potential insights for predicting future growth trends, with implications for medical and health management.

4.2 CD4 data

CD4 cells play a vital role in immune function, since CD4 counts on average decrease throughout the disease incubation period and disease progression can be assessed by measuring the number of CD4 cells. This dataset contains multi-center AIDS Cohort Study providing a total of 2376 CD4+ cell counts of 369 HIV-infected men covering a period of approximately eight and a half years. The number of measurements for each individual varies from 1 to 12. This dataset can be seen as an irregular sparse dataset.

Since we do not have true labels, we use the bootstrap technique to evaluate the stability of the clustering results, the main goal of which is to quantify the reliability of different parameter configurations and help users determine whether the clustering structure is statistically meaningful, and then select the optimal number of clusters. The stability of clustering is commonly assessed through resampling schemes, with the resulting stability score used to determine the number of clusters (Monti et al., 2003; Lange et al., 2004). Following the approach of Hennig (2007), we employ the Jaccard coefficient (Jaccard, 1901) to quantify similarity. Specifically, for each cluster in the original partition, we identify its most similar counterpart in a bootstrap-resampled clustering and compute their Jaccard similarity. Cluster-wise stability is then defined as the mean similarity aggregated across bootstrap replicates. A score exceeding 0.6 generally indicates a significant clustering structure. An evaluation of cluster stability across different solutions (Supplementary Figure S.3) shows that the 4-cluster demonstrates the most favorable profile. The 2-cluster solution is imbalanced, and the 3- and 5-cluster solutions yield low stability scores. However, in the 4-cluster solution, three clusters achieve stability scores near 0.6, supporting its selection as the most structurally meaningful.

Then we perform TSFDSC to cluster the CD4 cells and observe the curves in each of 4 clusters, as shown in Figure 4. Cluster 1 has a flat "U" shape while cluster 2 has

a flat inverted "U" shape, and its CD4 cell counts remain relatively stable for several years around the year of seroconversion and vary significantly only in the years away from seroconversion. For clusters 3 and 4, their CD4 cell counts have remarkable changes near the year of seroconversion, and also fluctuate away from that year, showing a flat "W" shape and a flat "M" shape respectively. Although there are some exceptions, the shape of these curves is generally consistent with the decline in CD4 cell counts after the confirmation of AIDS.

5. Discussion

Rooted in spectral graph theory, TSFDSC integrates spectral analysis with functional clustering, preserving temporal continuity and uncovering underlying spectral features. This approach excels at clustering within arbitrarily shaped sample spaces and converging toward globally optimal solutions. It is particularly effective for categorizing samples with numerical variations or specific shape characteristics and demonstrates robustness in handling data with sparsity or varying sampling time points. Results from our simulated and real data examples show that TSFDSC excels at identifying clusters that capture critical differences between groups of individuals. By offering an in-depth analysis of the intrinsic nature of functional data, TSFDSC addresses the limitations of traditional clustering techniques. Its key strengths lie in managing complex dynamic changes and incorporating spectral analysis, making it an ideal choice for applications requiring precise and nuanced clustering in diverse data contexts.

Supplementary Materials

The supplementary file contains the proofs of Theorems 2.1-2.2 and more extensive numerical results.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (12271255, 12371267), Qing Lan Project of Jiangsu Province, Postgraduate Research and Practice Innovation Program of Jiangsu Province (KYCX25_2447), the National Statistical Science Research Project of China (2024LY059), project of Joint Lab for Statistics and Finance, and Jiangsu Provincial Key Discipline Construction Project (Statistics).

References

Abraham, C., P. A. Cornillon, E. Matzner-Løber, and N. Molinari (2003). Unsupervised curve clustering using B-splines. *Scandinavian Journal of Statistics 30*(3), 581–595.

Aggarwal, C. C. and C. K. Reddy (2018). *Data Clustering*. Chapman and Hall/CRC, New York.

Al Alawi, M. (2021). *Spectral clustering and downsampling-based model selection for functional data*. Ph. D. thesis, University of Glasgow.

Bogin, B. (1999). Evolutionary perspective on human growth. *Annual Review of Anthropology 28*(1), 109–153.

Cao, G., L. Wang, Y. Li, and L. Yang (2016). Oracle-efficient confidence envelopes for covariance functions in dense functional data. *Statistica Sinica 26*(1), 359–383.

Chen, F., G. Cheung, and X. Zhang (2024). Manifold graph signal restoration using gradient graph Laplacian regularizer. *IEEE Transactions on Signal Processing 72*, 744–761.

Chiou, J.-M. and P.-L. Li (2007). Functional clustering and identifying substructures of longitudinal data. *Journal of the Royal Statistical Society Series B: Statistical Methodology 69*(4), 679–699.

Chiou, J.-M. and P.-L. Li (2008). Correlation-based functional clustering via subspace projection. *Journal of the American Statistical Association 103*(484), 1684–1692.

De Boor, C. (1978). *A Practical Guide to Splines*. Springer, New York.

Delaigle, A., P. Hall, and T. Pham (2019). Clustering functional data into groups by using projections. *Journal*

of the Royal Statistical Society Series B: Statistical Methodology 81(2), 271–304.

Fan, J. and I. Gijbels (1996). *Local Polynomial Modelling and Its Applications*. Routledge, New York.

Floriello, D. and V. Vitelli (2017). Sparse clustering of functional data. *Journal of Multivariate Analysis 154*, 1–18.

Garcia-Escudero, L. A. and A. Gordaliza (2005). A proposal for robust curve clustering. *Journal of Classification 22*(2), 185–201.

Grone, R., R. Merris, and V. S. Sunder (1990). The Laplacian spectrum of a graph. *SIAM Journal on Matrix Analysis and Applications 11*(2), 218–238.

Hennig, C. (2007). Cluster-wise assessment of cluster stability. *Computational Statistics & Data Analysis 52*(1), 258–271.

Huang, W. and D. Ruppert (2023). Copula-based functional bayes classification with principal components and partial least squares. *Statistica Sinica 33*(1), 55–84.

Ieva, F., A. M. Paganoni, D. Pigoli, and V. Vitelli (2013). Multivariate functional clustering for the morphological analysis of electrocardiograph curves. *Journal of the Royal Statistical Society: Series C (Applied Statistics) 62*(3), 401–418.

Jaccard, P. (1901). Distribution de la flore alpine dans le bassin des dranses et dans quelques régions voisines. *Bulletin de la Société Vaudoise des Sciences Naturelles 37*, 241–272.

Jacques, J. and C. Preda (2014). Functional data clustering: a survey. *Advances in Data Analysis and Classification 8*, 231–255.

James, G. M. and C. A. Sugar (2003). Clustering for sparsely sampled functional data. *Journal of the American Statistical Association 98*(462), 397–408.

Kachouie, N. N. and M. Shutaywi (2020). Weighted mutual information for aggregated kernel clustering. *Entropy 22*(3), 351.

Kronenberg, H. M. (2007). *Williams Textbook of Endocrinology E-Book*. Elsevier Health Sciences.

Lange, T., V. Roth, M. L. Braun, and J. M. Buhmann (2004). Stability-based validation of clustering solutions. *Neural Computation 16*(6), 1299–1323.

Largo, R., T. Gasser, A. Prader, W. Stuetzle, and P. Huber (1978). Analysis of the adolescent growth spurt using smoothing spline functions. *Annals of Human Biology 5*(5), 421–434.

Li, J. and L. Yang (2023). Statistical inference for functional time series. *Statistica Sinica 33*(1), 519–549.

Li, Y. and T. Hsing (2010). Uniform convergence rates for nonparametric regression and principal component analysis in functional/longitudinal data. *The Annals of Statistics 38*(6), 3321–3351.

Liang, D., H. Huang, Y. Guan, and F. Yao (2023). Test of weak separability for spatially stationary functional field. *Journal of the American Statistical Association 118*(543), 1606–1619.

Ma, S. (2014). A plug-in the number of knots selector for polynomial spline regression. *Journal of Nonparametric Statistics 26*(3), 489–507.

Ma, T., F. Yao, and Z. Zhou (2024). Network-level traffic flow prediction: Functional time series vs. functional neural network approach. *The Annals of Applied Statistics 18*(1), 424–444.

Monti, S., P. Tamayo, J. Mesirov, and T. Golub (2003). Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data. *Machine Learning 52*, 91–118.

Ramsay, J. (2024). *fda: Functional Data Analysis*. R package version 6.1.9.

Ramsay, J. and B. Silverman (2005). *Functional Data Analysis*. Springer, New York.

Von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and Computing 17*, 395–416.

Von Luxburg, U., M. Belkin, and O. Bousquet (2008). Consistency of spectral clustering. *Annals of Statistics 36*(2), 555–586.

Wang, J., G. Cao, L. Wang, and L. Yang (2020). Simultaneous confidence band for stationary covariance function of dense functional data. *Journal of Multivariate Analysis 176*, 104584.

Wang, J., L. Gu, and L. Yang (2022). Oracle-efficient estimation for functional data error distribution with simultaneous confidence band. *Computational Statistics & Data Analysis 167*, 107363.

Wang, J. L., J. M. Chiou, and H. G. Müller (2016). Functional data analysis. *Annual Review of Statistics and Its Application 3*(1), 257–295.

Xue, K., J. Yang, and F. Yao (2024). Optimal linear discriminant analysis for high-dimensional functional data. *Journal of the American Statistical Association 119*(546), 1055–1064.

Yao, F., H. G. Müller, and J. L. Wang (2005). Functional data analysis for sparse longitudinal data. *Journal of the American Statistical Association 100*(470), 577–590.

Zhang, M. and A. Parnell (2023). Review of clustering methods for functional data. *ACM Transactions on Knowledge Discovery from Data 17*(7), 1–34.

Zhang, X. and J.-L. Wang (2016). From sparse to dense functional data and beyond. *The Annals of Statistics 44*(5), 2281–2321.

Zhou, H., F. Yao, and H. Zhang (2023). Functional linear regression for discretely observed data: from ideal to reality. *Biometrika 110*(2), 381–393.

Table 1: Simulation scenario settings, where $(var(\xi_{i1}^{(j)}), var(\xi_{i2}^{(j)}))^\top = \zeta^{(j)}$ for $j = 1, 2, 3$ and $i = 1, \ldots, n$.

Scenario		curve	σ
1	a	$Y_{ij}^{(1)} = \mu_2(t_{ij}) + \sum_{k=1}^{2} \xi_{ik}^{(1)} \phi_k^{(1)}(t_{ij}) + \epsilon_{ij}^{(1)}$	0.01
	b	$Y_{ij}^{(2)} = \mu_2(t_{ij}) + \sum_{k=1}^{2} \xi_{ik}^{(2)} \phi_k^{(2)}(t_{ij}) + \epsilon_{ij}^{(2)}$	
	c	$Y_{ij}^{(3)} = \mu_2(t_{ij}) + \sum_{k=1}^{2} \xi_{ik}^{(3)} \phi_k^{(3)}(t_{ij}) + \epsilon_{ij}^{(3)}$	
2	a	$Y_{ij}^{(1)} = \mu_2(t_{ij}) + \sum_{k=1}^{2} \xi_{ik}^{(1)} \phi_k^{(1)}(t_{ij}) + \epsilon_{ij}^{(1)}$	0.3
	b	$Y_{ij}^{(2)} = \mu_2(t_{ij}) + \sum_{k=1}^{2} \xi_{ik}^{(2)} \phi_k^{(2)}(t_{ij}) + \epsilon_{ij}^{(2)}$	
	c	$Y_{ij}^{(3)} = \mu_2(t_{ij}) + \sum_{k=1}^{2} \xi_{ik}^{(3)} \phi_k^{(3)}(t_{ij}) + \epsilon_{ij}^{(3)}$	
3	a	$Y_{ij}^{(1)} = \mu_2(t_{ij}) + 1 + \sum_{k=1}^{2} \xi_{ik}^{(1)} \phi_k^{(2)}(t_{ij}) + \epsilon_{ij}^{(1)}$	0.01
	b	$Y_{ij}^{(2)} = \mu_2(t_{ij}) - 1 + \sum_{k=1}^{2} \xi_{ik}^{(1)} \phi_k^{(2)}(t_{ij}) + \epsilon_{ij}^{(2)}$	
	c	$Y_{ij}^{(3)} = \mu_2(t_{ij}) + \sum_{k=1}^{2} \xi_{ik}^{(1)} \phi_k^{(2)}(t_{ij}) + \epsilon_{ij}^{(3)}$	
4	a	$Y_{ij}^{(1)} = \mu_2(t_{ij}) + 1 + \sum_{k=1}^{2} \xi_{ik}^{(1)} \phi_k^{(2)}(t_{ij}) + \epsilon_{ij}^{(1)}$	0.3
	b	$Y_{ij}^{(2)} = \mu_2(t_{ij}) - 1 + \sum_{k=1}^{2} \xi_{ik}^{(1)} \phi_k^{(2)}(t_{ij}) + \epsilon_{ij}^{(2)}$	
	c	$Y_{ij}^{(3)} = \mu_2(t_{ij}i) + \sum_{k=1}^{2} \xi_{ik}^{(1)} \phi_k^{(2)}(t_{ij}) + \epsilon_{ij}^{(3)}$	
5	a	$Y_{ij}^{(1)} = \mu_1(t_{ij}) + \sum_{k=1}^{2} \xi_{ik}^{(1)} \phi_k^{(2)}(t_{ij}) + \epsilon_{ij}^{(1)}$	0.01
	b	$Y_{ij}^{(2)} = \mu_2(t_{ij}) + \sum_{k=1}^{2} \xi_{ik}^{(1)} \phi_k^{(2)}(t_{ij}) + \epsilon_{ij}^{(2)}$	
	c	$Y_{ij}^{(3)} = \mu_3(t_{ij}) + \sum_{k=1}^{2} \xi_{ik}^{(1)} \phi_k^{(2)}(t_{ij}) + \epsilon_{ij}^{(3)}$	
6	a	$Y_{ij}^{(1)} = \mu_1(t_{ij}) + \sum_{k=1}^{2} \xi_{ik}^{(1)} \phi_k^{(2)}(t_{ij}) + \epsilon_{ij}^{(1)}$	0.3
	b	$Y_{ij}^{(2)} = \mu_2(t_{ij}) + \sum_{k=1}^{2} \xi_{ik}^{(1)} \phi_k^{(2)}(t_{ij}) + \epsilon_{ij}^{(2)}$	
	c	$Y_{ij}^{(3)} = \mu_3(t_{ij}) + \sum_{k=1}^{2} \xi_{ik}^{(1)} \phi_k^{(2)}(t_{ij}) + \epsilon_{ij}^{(3)}$	

Table 2: Clustering results (part) for settings 1.a with empirical formula-based N_s over 100 replications (incorporate comparisons with FSC-S and kCFC). The best performance in each simulation scenario is indicated with an asterisk "*".

m_i	Scenario	n = 50				n = 200			
		FD k-means	TSFDSC	FSC-S(D_0)	kCFC	FD k-means	TSFDSC	FSC-S(D_0)	kCFC
6	1	0.1319*	0.1308	-	-	0.1389*	0.1129	-	-
	2	0.0623	0.0818*	-	-	0.0400	0.0696*	-	-
	3	0.7699	0.9465*	-	0.4419	0.7623	0.9488*	-	0.0821
	4	0.7107	0.8041*	-	0.4136	0.7107	0.8776*	-	0.0606
	5	0.9518*	0.9295	-	0.9213	0.9479	0.9747	-	0.7823
	6	0.9303	0.9336*	-	0.8842	0.9279	0.9547*	-	0.7370
21	1	0.1318*	0.1258	0.1049	-	0.1335*	0.1115	0.0788	-
	2	0.0329	0.0344	0.0719*	-	0.0183	0.0311	0.0436*	-
	3	0.7695	0.9539*	0.6388	0.3155	0.7625	0.9315*	0.6286	0.0272
	4	0.7430	0.7855*	0.6217	0.3252	0.7402	0.8411*	0.0036	0.0549
	5	0.9502*	0.9343	0.9391	0.9055	0.9471	0.9621*	0.9361	0.7084
	6	0.9427*	0.9388	0.9321	0.9042	0.9398	0.9428*	0.9245	0.7006
101	1	0.1309*	0.1264	0.1047	-	0.1377*	0.1009	0.0798	-
	2	0.1301	0.1401*	0.0959	-	0.1305	0.1410*	0.0710	-
	3	0.7701	0.9386*	0.6376	0.2391	0.7627	0.9568*	0.6274	0.0338
	4	0.7652	0.9424*	0.6353	0.2422	0.7564	0.9519*	0.6229	0.0485
	5	0.9505	0.9551*	0.9387	0.9154	0.9467	0.9705*	0.9346	0.7139
	6	0.9481	0.9540*	0.9372	0.9131	0.9455*	0.9363	0.9318	0.6953

Table 3: Clustering results (part) for settings 1.a and 2.a with empirical formula-based N_s over 100 replications.

m_i	Scenario	n				
		50				
		FD k-means	TSFDSC			
			ι			
			adaptive	0.1	0.5	0.9
6	1	0.1319	0.1308	0.1029	0.1451	0.1522*
	2	0.0623	0.0818*	0.0331	0.0676	0.0809
	3	0.7699	0.9465*	0.7636	0.9159	0.9362
	4	0.7107	0.8041	0.7041	0.7946	0.8094*
	5	0.9518*	0.9295	0.9443	0.9505	0.9259
	6	0.9303	0.9336	0.9240	0.9473*	0.9379
21	1	0.1318	0.1258	0.0990	0.1419	0.1526*
	2	0.0329	0.0344	0.0239	0.0356	0.0425*
	3	0.7695	0.9539*	0.7620	0.9080	0.9413
	4	0.7430	0.7855	0.7321	0.8282	0.8381*
	5	0.9502*	0.9343	0.9307	0.9460	0.9213
	6	0.9427*	0.9388	0.9296	0.9405	0.9370
101	1	0.1309	0.1264	0.1041	0.1409	0.1529*
	2	0.1301	0.1401*	0.0981	0.1368	0.1356
	3	0.7701	0.9386*	0.7632	0.9194	0.9361
	4	0.7652	0.9424*	0.7566	0.9062	0.9412
	5	0.9505	0.9551*	0.9398	0.9379	0.9424
	6	0.9481	0.9540*	0.9388	0.9334	0.9174

Table 4: Clustering results for settings 1.b and 2.b, with empirical formula-based N_s over 100 replications.

m_i	Scenario	n					
		50		100		200	
		FD k-means	TSFDSC	FD k-means	TSFDSC	FD k-means	TSFDSC
4-11	1	0.1411*	0.1319	0.1383*	0.1190	0.1503*	0.1075
	2	0.0736	0.0769*	0.0521	0.0749*	0.0529	0.0808*
	3	0.7638	0.8784*	0.7620	0.9278*	0.7514	0.9402*
	4	0.7188	0.8375*	0.7046	0.8374*	0.7083	0.8822*
	5	0.8553*	0.7986	0.8963*	0.8492	0.8920*	0.7976
	6	0.8485	0.8881*	0.8904*	0.8691	0.8847*	0.8751
4-21	1	0.1360*	0.1248*	0.1406*	0.1180	0.1398*	0.1078
	2	0.0328	0.0396*	0.0193	0.0293*	0.0143*	0.0135
	3	0.7659	0.9054*	0.7660	0.9601*	0.7548	0.9851*
	4	0.7246	0.7259*	0.7146*	0.6873	0.7262*	0.6712
	5	0.8712	0.9162*	0.8658	0.8979*	0.8829*	0.8749
	6	0.8494	0.8887*	0.8701	0.9023*	0.8844*	0.8778
21-101	1	0.1335*	0.1303	0.1318*	0.1197	0.1316*	0.1027
	2	0.1206*	0.1184	0.1243	0.1368*	0.1167	0.1316*
	3	0.7691	0.9399*	0.7771	0.9469*	0.7522	0.9137*
	4	0.7675	0.9050*	0.7659	0.9245*	0.7486	0.9435*
	5	0.8790	0.9282*	0.8781	0.9488*	0.8751	0.8989*
	6	0.8734	0.9398*	0.8785	0.9629*	0.8840	0.9445*

Table 5: Clustering results for regular sparse settings over 100 replications.

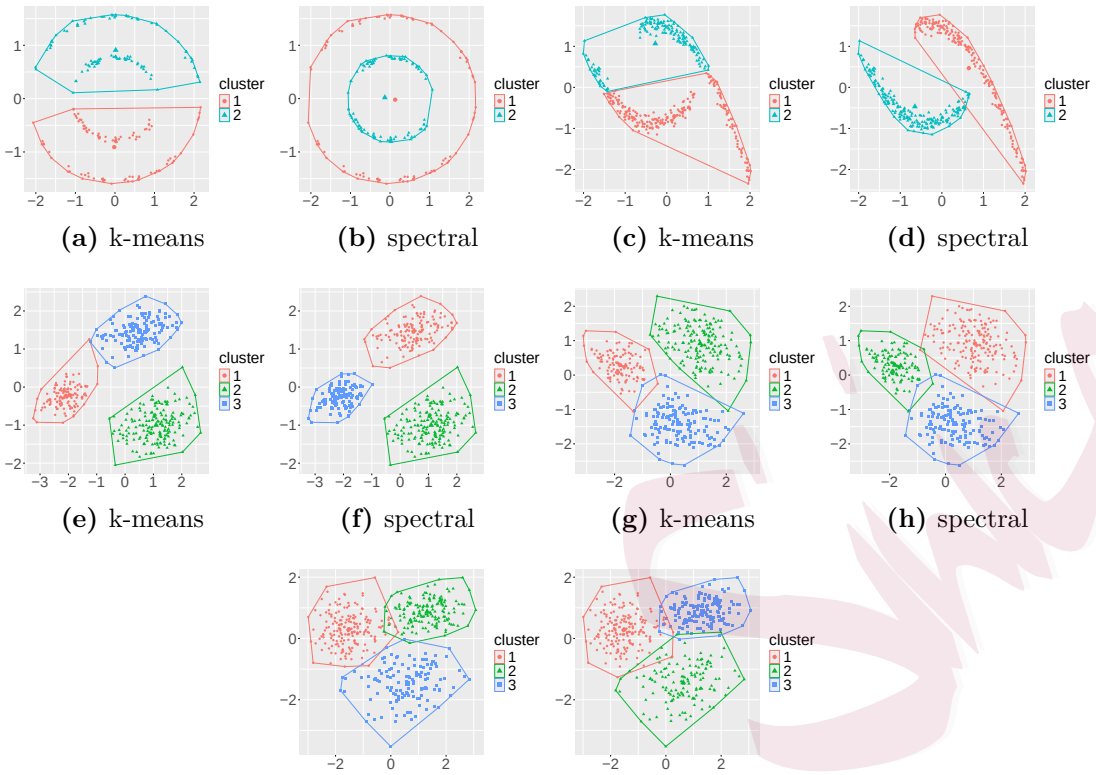
m_i	Scenario	n = 50		100		200	
		FD k-means	TSFDSC	FD k-means	TSFDSC	FD k-means	TSFDSC
6	1	0.1409*	0.1287	0.1418*	0.1106	0.1584*	0.1011
	2	0.0738	0.0840*	0.0607	0.0772*	0.0610	0.0752*
	3	0.6666	0.9092*	0.6657	0.9452*	0.6588	0.9517*
	4	0.6270	0.6616*	0.6269	0.7073*	0.6199	0.7578*
	5	0.9179	0.9287*	0.9185	0.9574*	0.9113	0.9410*
	6	0.8888	0.9298*	0.8772	0.9209*	0.8753	0.9438*
11	1	0.1303*	0.1283	0.1304*	0.1238	0.1498*	0.1126
	2	0.1064	0.1071*	0.0978	0.1018*	0.1062*	0.0911
	3	0.6601	0.9129*	0.6671	0.9515*	0.6555	0.9513*
	4	0.6372	0.7245*	0.6388	0.8002*	0.6341	0.8351*
	5	0.8626	0.9425*	0.8758	0.9616*	0.8709	0.9495*
	6	0.9129	0.9176*	0.9126	0.9492*	0.9082	0.9504*
4-11	1	0.1447*	0.1410	0.1578*	0.1171	0.1573*	0.1127
	2	0.0764	0.0841*	0.0638	0.0693*	0.0768	0.0812*
	3	0.6557	0.8579*	0.6530	0.9183*	0.6459	0.9542*
	4	0.6324	0.6999*	0.6079	0.7228*	0.6162	0.7665*
	5	0.9128*	0.9124	0.9269	0.9645*	0.9185	0.9569*
	6	0.8803*	0.8757	0.8899	0.8954*	0.8820	0.9016*
4-21	1	0.1375*	0.1338	0.1491*	0.1148	0.1506*	0.1054
	2	0.0874	0.0916*	0.0926*	0.0889	0.0898*	0.0783
	3	0.6548	0.8511*	0.6552	0.9323*	0.6475	0.9037*
	4	0.6076	0.6355*	0.5901	0.5980*	0.6112	0.7284*
	5	0.8905	0.9062*	0.9053	0.9076*	0.8950	0.8984*
	6	0.8416	0.8614*	0.8770	0.8811*	0.8792	0.8918*

Table 6: Clustering results for setting 3 over 100 replications.

m_i	Scenario	n = 50		100		200	
		FD k-means	TSFDSC	FD k-means	TSFDSC	FD k-means	TSFDSC
1-4	1	0.0678(0.4544)*	0.0635(0.4416)	0.0680(0.4562)*	0.0609(0.4419)	0.0719(0.4591)*	0.0586(0.4356)
	2	0.0150(0.3631)	0.0215(0.3972)*	0.0123(0.3652)	0.0168(0.3909)*	0.0077(0.3578)	0.0127(0.3787)*
	3	0.4897(0.7865)*	0.4582(0.7472)	0.4794(0.7819)*	0.4605(0.7634)	0.4826(0.7912)*	0.4538(0.7612)
	4	0.3112(0.6396)	0.3504(0.6918)*	0.2860(0.6139)	0.3313(0.6718)*	0.2518(0.5916)	0.2856(0.6364)*
	5	0.4261(0.7376)*	0.3995(0.7029)	0.4341(0.7551)*	0.4034(0.7185)	0.4355(0.7577)*	0.4361(0.7528)
	6	0.2704(0.6013)	0.3106(0.6581)*	0.2521(0.5902)	0.3103(0.6606)*	0.2529(0.5961)	0.2756(0.6315)*

Table 7: Clustering results of height curves and growth curves.

height curves clustering					
		cluster 1	cluster 2	NMI	Ac
FD k-means	male	32	7	0.38	0.85
	female	7	47		
TSFDSC(Bs)	male	39	0	0.78	0.96
	female	4	50		
TSFDSC(PACE)	male	6	33	0.05	0.63
	female	1	53		
k-means(cFPC)	male	23	16	0.06	0.65
	female	17	37		
SC(cFPC)	male	15	24	0.15	0.72
	female	2	52		
Delaigle et al. (2019)	-	-	-	-	0.93
Al Alawi (2021)	-	-	-	-	0.94
Chiou and Li (2007)	-	-	-	-	0.94
growth curves clustering					
		cluster 1	cluster 2	NMI	Ac
FD k-means	male	38	1	0.55	0.89
	female	9	45		
TSFDSC(Bs)	male	38	1	0.58	0.90
	female	8	46		
TSFDSC(PACE)	male	23	16	0.41	0.83
	female	0	54		
k-means(cFPC)	male	38	1	0.55	0.89
	female	9	45		
SC(cFPC)	male	23	16	0.41	0.83
	female	0	54		
Floriello and Vitelli (2017)	male	37	2	0.50	0.88
	female	9	45		

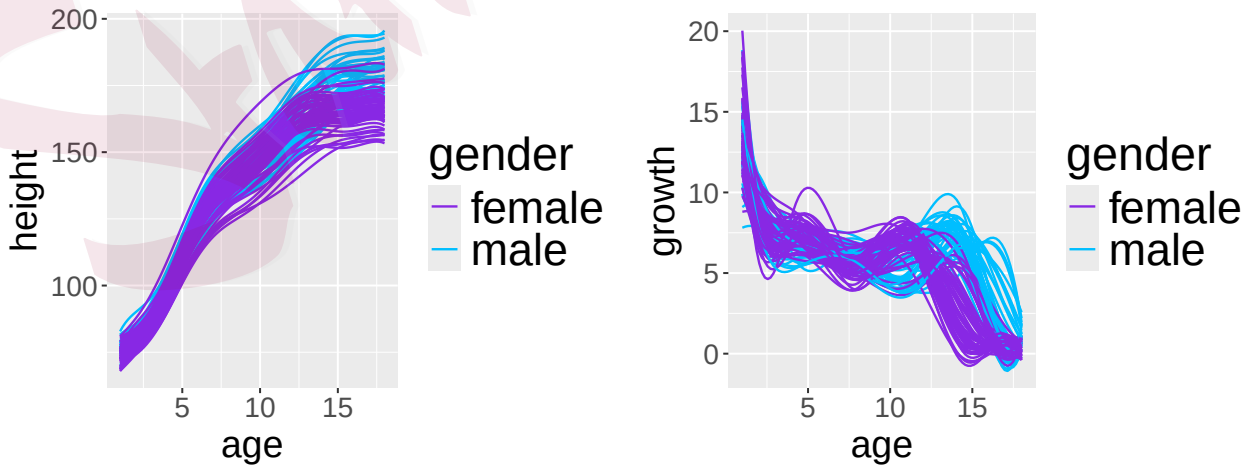


(a) k-means (b) spectral (c) k-means (d) spectral

(e) k-means (f) spectral (g) k-means (h) spectral

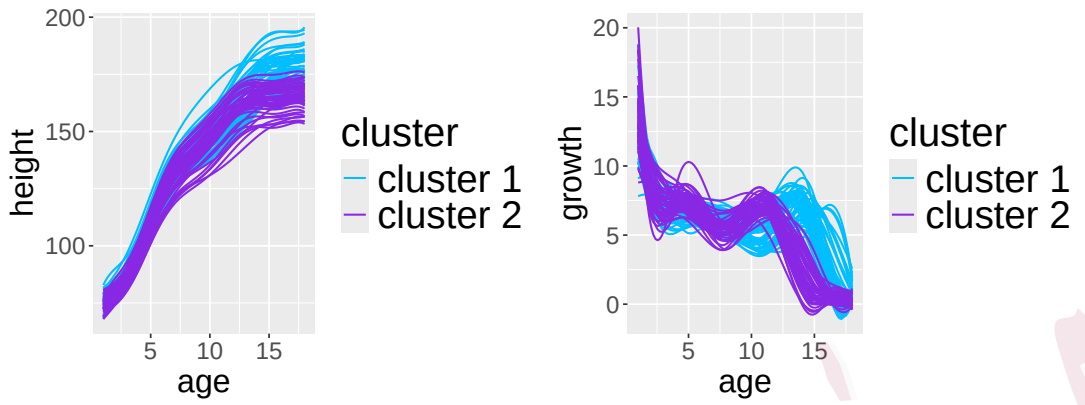
(i) k-means (j) spectral

Figure 1: Plots of clusters: Panels (a) and (b) illustrate k-means and spectral clustering for two types of circular data ($x^2 + y^2 = 1$ and $x^2 + y^2 = 4$), Panels (c) and (d) display clusters for two different curve datasets ($y = x^2$ and $y = -(x-1)^2+2$), Panels (e)-(j) show clusters from randomly generated data (using genRandomClust in R). Specifically, panels (e) and (f) depict well-separated clusters, (g)-(h) show a separated cluster structure, and (i)-(j) illustrate compact clusters.



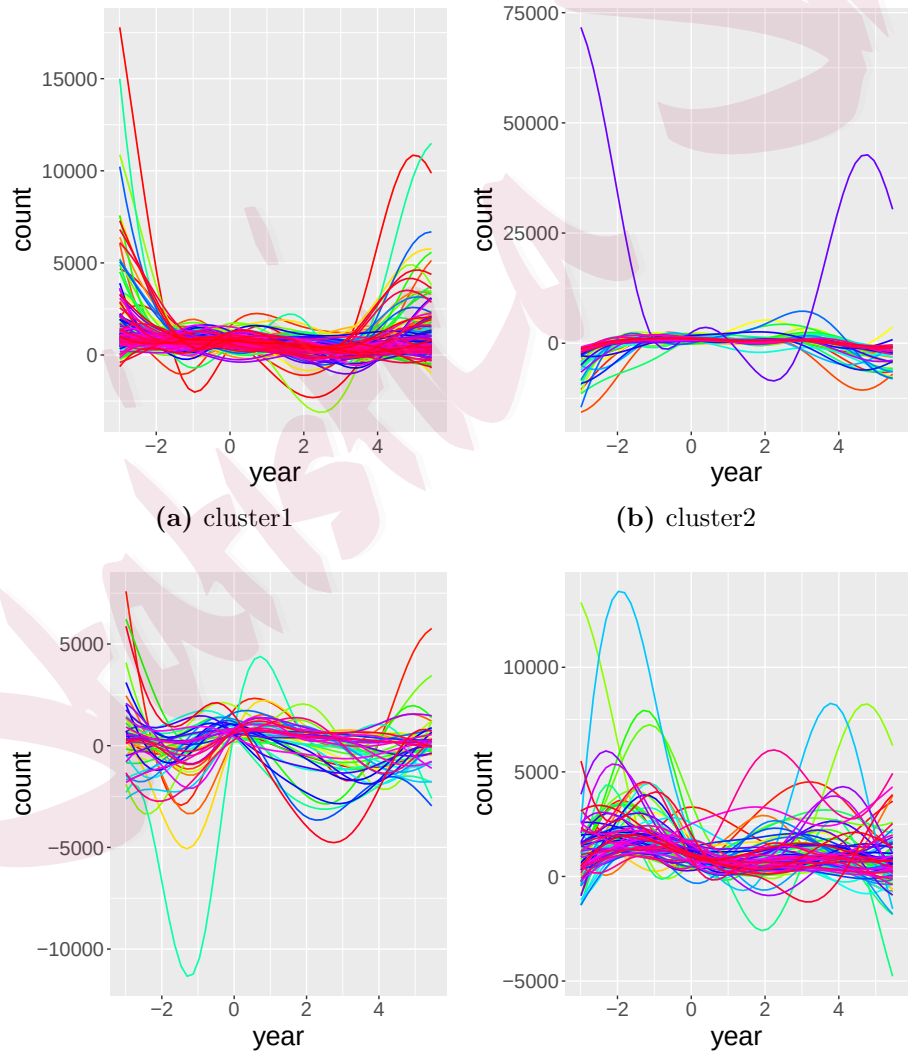
(a) height curves **(b)** growth curves

Figure 2: B-spline smoothing for Berkeley Growth data



(a) height curves **(b)** growth curves

Figure 3: Clustering of Berkeley Growth based on TSFDSC



(a) cluster1 **(b)** cluster2

(c) cluster3 **(d)** cluster4

Figure 4: Fitted sample curve plot based on the results from PACE